DISSERTATION

ALGORITHMS FOR FEATURE SELECTION AND PATTERN RECOGNITION ON GRASSMANN MANIFOLDS

Submitted by

Sofya Chepushtanova

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2015

Doctoral Committee:

Advisor: Michael Kirby

Chris Peterson
Dan Bates
Asa Ben-Hur

ABSTRACT

ALGORITHMS FOR FEATURE SELECTION AND PATTERN RECOGNITION ON GRASSMANN MANIFOLDS

This dissertation presents three distinct application-driven research projects united by ideas and topics from geometric data analysis, optimization, computational topology, and machine learning.

We first consider hyperspectral band selection problem solved by using sparse support vector machines (SSVMs). A supervised embedded approach is proposed using the property of SSVMs to exhibit a model structure that includes a clearly identifiable gap between zero and non-zero feature vector weights that permits important bands to be definitively selected in conjunction with the classification problem. An SSVM is trained using bootstrap aggregating to obtain a sample of SSVM models to reduce variability in the band selection process. This preliminary sample approach for band selection is followed by a secondary band selection which involves retraining the SSVM to further reduce the set of bands retained. We propose and compare three adaptations of the SSVM band selection algorithm for the multiclass problem. We illustrate the performance of these methods on two benchmark hyperspectral data sets.

Second, we propose an approach for capturing the signal variability in data using the framework of the Grassmann manifold (Grassmannian). Labeled points from each class are sampled and used to form abstract points on the Grassmannian. The resulting points have representations as orthonormal matrices and as such do not reside in Euclidean space in the usual sense. There are a variety of metrics which allow us to determine distance matrices that can be used to realize the Grassmannian as an embedding in Euclidean space.

Multidimensional scaling (MDS) determines a low dimensional Euclidean embedding of the manifold, preserving or approximating the Grassmannian geometry based on the distance measure. We illustrate that we can achieve an isometric embedding of the Grassmann manifold using the chordal metric while this is not the case with other distances. However, non-isometric embeddings generated by using the smallest principal angle pseudometric on the Grassmannian lead to the best classification results: we observe that as the dimension of the Grassmannian grows, the accuracy of the classification grows to 100% in binary classification experiments. To build a classification model, we use SSVMs to perform simultaneous dimension selection. The resulting classifier selects a subset of dimensions of the embedding without loss in classification performance.

Lastly, we present an application of persistent homology to the detection of chemical plumes in hyperspectral movies. The pixels of the raw hyperspectral data cubes are mapped to the geometric framework of the Grassmann manifold where they are analyzed, contrasting our approach with the more standard framework in Euclidean space. An advantage of this approach is that it allows the time slices in a hyperspectral movie to be collapsed to a sequence of points in such a way that some of the key structure within and between the slices is encoded by the points on the Grassmannian. This motivates the search for topological structure, associated with the evolution of the frames of a hyperspectral movie, within the corresponding points on the manifold. The proposed framework affords the processing of large data sets, such as the hyperspectral movies explored in this investigation, while retaining valuable discriminative information. For a particular choice of a distance metric on the Grassmannian, it is possible to generate topological signals that capture changes in the scene after a chemical release.

ACKNOWLEDGEMENTS

I would like to thank my advisor Michael Kirby for his guidance, encouragement, invaluable insight and feedback, and for the opportunity to be involved in exciting research projects.

Many thanks as well to Chris Peterson for bringing his views and ideas, and especially for his insights into persistent homology theory and applications.

I am very grateful for the collaboration and friendship I have had in the Department of Mathematics, and, particularly, in the Pattern Analysis Lab (PAL). Thank you Lori Ziegelmeier for being my research and travel companion. Thank you Tim Marrinan, Drew Schwickerath, and Tegan Emerson for being helpful resources and great pals.

I also thank Henry Adams for providing additional MATLAB software for determining the indices of connected components in persistence homology barcodes generated by JavaPlex.

Finally, I am deeply thankful to the support of my family. Thank you mom, dad, and Valeriy. And, of course, I thank my husband, Alexei, and my little daughters, Anna and Yana, for their love, patience, and encouragement throughout my PhD journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1. Overview

Nowadays it has become possible to acquire large and information-rich data sets for different applications. There are many difficulties associated with understanding such data sets, e.g., the data may be incomplete, noisy and have thousands of features. Consider, for instance, the task of predicting a diagnosis or treatment for patients by analyzing their gene expression data. The presence of a large collection of irrelevant features in the numerous measurements of gene expression just add to the computational complexity, without helping much to build a prediction model. Another example of high-dimensional data is hyperspectral data. Hyperspectral imagery collects data as a set of images simultaneously in tens to hundreds of narrow wavelength bands, forming three-dimensional data cubes [1]. Each pixel can be represented as a vector in $\mathbb{R}^n$, where n is a typically large number of spectral wavelength bands. Rich information contained in hyperspectral data can be useful for different tasks, but some information can be noisy and redundant.

Thus, we are often interested in obtaining reduced data representation, efficient for a particular prediction task or data visualization [2]. Some large data sets may require a form of compression that retains their geometric structure. The goal of this dissertation is to introduce some novel data analysis techniques and frameworks based on tools from geometric and topological data analysis and machine learning. Below we briefly discuss three approaches devoted to problems of dimensionality reduction and pattern recognition in some challenging applications.

Dimensionality reduction can be done using feature extraction or feature selection techniques. *Feature extraction* transforms the data to a lower dimensional space. In the last decade there has been a number of fundamental contributions to this problem of geometric data reduction, including ISOMAP, Locally-Linear Embedding (LLE), Laplacian Eigenmaps, and Maximum Variance Unfolding (MVU). The proof of Whitney's easy embedding theorem has led to a framework for constructing Bilipschitz mappings for dimension preserving data reduction [3]. *Feature selection* is the process of selecting a relevant set of features while maintaining or improving the performance of a prediction model. There exists a variety of feature selecting techniques that are categorized into filters, wrappers, and embedded algorithms [4]. The last group of methods perform feature selection as part of the model construction process. For instance, learning with sparsity-inducing norms in the context of linear or logistic regression or support vector machines (SVMs) [5] drives many redundant feature weights to zero [6].

We use a sparsity promoting approach as a solution to the hyperspectral band selection problem [7]. Before introducing our method, in Chapter 2 we discuss sparse SVM classifiers (SSVMs) that simultaneously classify and automatically select features in the input space, therefore reducing its dimension. We formulate and discuss the difference between standard and sparse SVMs, and introduce the primal dual interior point method as a solver for SSVMs.

After this, in Chapter 3, we propose a hyperspectral band selection algorithm based on the feature selection property of SSVMs. We introduce the band selection problem and make an overview of the related methods in the literature. Our embedded supervised approach contains two main steps, namely, variability reduction and final ratio-based selection. The 2-class verison of the algorithm is further extended to the multiclass case. Our results on

two hyperspectral data sets show the effectiveness of this methodology used both separately and in combination with other band selection strategies.

In Chapter 4, we propose a geometric approach for capturing the variability in hyperspectral data using the framework of the Grassmann manifold (Grassmannian) to perform set-to-set pattern recognition. The Grassmannian can be interpreted as a linear span of a set of data samples [8]. Original data points are organized as subspaces (abstract points) on the Grassmannian, and then embedded into Euclidean space, where an SSVM is trained to perform classification. The SSVM selects a subset of optimal embedding dimensions, which can be used for improving classification rates or embedding visualization. The proposed framework results in classification accuracy that grows up to 100% in binary classification experiments, including high difficulty classification cases. The method is extended it to the multiclass case, and embeddings obtained under different distance measures on the manifold are compared and analyzed for isometry.

The Grassmannian framework affords a form of data compression while retaining data structure. We propose using it in conjunction with a relatively new tool from topological data analysis (TDA), persistent homology (PH), based on building simplicial complexes on the data sets [9, 10]. PH has been used to find data structure in many applications in biology, computer graphics, and image processing. In Chapter 5, we explore uses of persistent homology for chemical plume detection in hyperspectral movies. We apply PH to hyperspectral data, encoded as abstract points on a Grassmann manifold which makes it feasible to analyze large volumes of hyperspectral data. Using PH as a multiscale method for determining the number of connected components in data, we capture the dynamical changes in a hyperspectal movie over time. The appropriate choice of a distance metric on the manifold results in generating strong topological signals.

Finally, Chapter 6 is devoted to conclusions of the dissertation and potential future work.

1.2. Definitions and Notation

We introduce our notation and definitions used in the dissertation.

- A vector in $\mathbb{R}^n$ is denoted with a bold lower case letter, e.g., $\mathbf{e}$ denotes a vector of all ones.
- A boldface capital letter denotes a matrix, e.g., $\mathbf{X} \in \mathbb{R}^{m \times n}$ is a real $m \times n$ matrix.
- The symbol $\doteq$ denotes a definition of the term to the left of the symbol by the expression to the right of the symbol.
- The ℓ_1, ℓ_2, and ℓ_∞-norms of a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ are defined as $\|\mathbf{x}\|_1 \doteq \sum_{i=1}^n |x_i|$, $\|\mathbf{x}\|_2 \doteq (\sum_{i=1}^n |x_i|^2)^{\frac{1}{2}}$, and $\|\mathbf{x}\|_\infty \doteq \max_i\{|x_i|\}$, respectively. The ℓ_p-norm is given by $\|\mathbf{x}\|_p \doteq (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$.
- For a general norm $\|\cdot\|$, the *dual norm* $\|\cdot\|'$ is defined as $\|\mathbf{x}\|' \doteq \max_{\|\mathbf{y}\|=1} \mathbf{x}^T\mathbf{y}$. Note that for $p, q > 1$, $1/p + 1/q = 1$, the ℓ_p-norm and ℓ_q-norm are dual. E.g., the ℓ_2-norm is dual to itself, and the ℓ_1-norm is dual to the ℓ_∞-norm.
- The *Frobenius norm* of a matrix $\mathbf{A}$ is given by $\|\mathbf{A}\|_F = \sqrt{\sum |\mathbf{A}_{ij}|^2} = \sqrt{\text{trace}(\mathbf{A}^T\mathbf{A})}$.
- Classification *accuracy* is the number of correct predictions made divided by the total number of predictions made.

CHAPTER 2

Linear SVM Classifiers

2.1. Introduction

This chapter provides background material on *linear* support vector machines (SVMs), with the emphasis on the ℓ_1-norm regularized SVM. The SVM is a state-of-the-art classification method[1] excellently described, for instance, in the book by Vapnik [5] or the tutorial by Burges [11]. SVMs fall into the general category of kernel methods, i.e., methods that depend on the data only through dot-products replaced with kernel (and, in general, non-linear) functions [12]. In this dissertation we consider linear SVM classifiers only, i.e., those with linear kernels, or simply, original dot-products.

The SVM is a robust supervised classification technique that has become the method of choice to solve difficult classification problems in a wide range of application domains such as bioinformatics [13], text classification [14], or hyperspectral remote sensing image classification [15]. We consider a class of the SVM classifiers that are based on ℓ_1-norm regularization, called *sparse* SVMs (SSVMs) [16–18]. The principal advantage of SSVMs is that, unlike ℓ_2-norm, or standard SVMs, they promote sparsity in the decision function, and therefore, reduce the input space dimension. We use SSVMs for hyperspectral band selection (Chapter 3) and for classification of data on embedded Grassmannian (Chapter 4), hence, it is important to understand the mechanism behind.

The chapter contains four sections: Section 2.2 on standard SVMs, Section 2.3 on sparse SVMs, Section 2.4 on the primal dual interior point algorithm used to solve SSVMs, and Section 2.5 containing a brief summary.

[1] The classification problem is the problem of determining which of several sets an object is a member of.

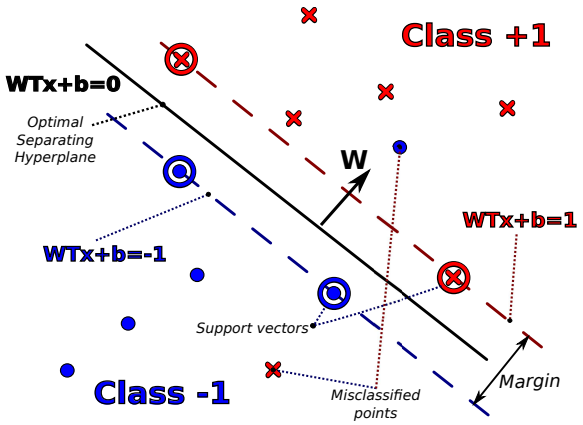


Figure 2.1: Separating hyperplane built by a binary SVM on non-separable data.

2.2. Standard SVMs

A standard (ℓ_2-norm) linear support vector machine (SVM) determines the optimal hyperplane $\{\mathbf{x} : \mathbf{x} \in \mathbb{R}^n, \mathbf{w}^T\mathbf{x} + b = 0\}$, maximally separating two classes of training data $\{\mathbf{x}_i, d_i\}$, $i = 1, \ldots, m$, where $d_i \in \{-1, +1\}$ are the class labels of the data points $\mathbf{x}_i \in \mathbb{R}^n$, $\mathbf{w}$ is the normal to the hyperplane, and b is the threshold [5, 11]. The class of a pattern $\mathbf{x}$ is predicted by $\text{sgn}(\mathbf{w}^T\mathbf{x} + b)$. Figure 2.1 shows the optimal hyperplane built by an SVM trained on non-separable data. The margin between classes is given by $2/\|\mathbf{w}\|_2$ [11].

To find the maximum margin hyperplane, one solves the following constrained optimization problem:

$$\begin{aligned} \underset{\mathbf{w},b,\boldsymbol{\xi}}{\text{minimize}} \quad & \frac{\|\mathbf{w}\|_2^2}{2} + C\mathbf{e}^T\boldsymbol{\xi} \\ \text{subject to} \quad & \mathbf{D}(\mathbf{X}\mathbf{w} + b\mathbf{e}) \geq \mathbf{e} - \boldsymbol{\xi}, \\ & \boldsymbol{\xi} \geq \mathbf{0}. \end{aligned} \tag{2.1}$$

Here $\mathbf{D}$ is the diagonal matrix with $\mathbf{D}_{ii} = d_i$, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^T$ is the training data matrix, $\boldsymbol{\xi}$ is an m-dimensional non-negative error slack variable, and C is a positive penalty parameter

that determines the trade-off between the SVM errors and the margin. The formulation (2.1) is also known as the soft-margin SVM [11]. The dual of (2.1) is

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \mathbf{e}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{D}\mathbf{X}\mathbf{X}^T\mathbf{D}\boldsymbol{\alpha} \\ \text{subject to} \quad & \mathbf{e}^T\mathbf{D}\boldsymbol{\alpha} = 0, \\ & \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{e}. \end{aligned} \tag{2.2}$$

Support vectors (SVs) are the data points that define the classifier, namely, those corresponding to the positive Lagrange multipliers $\alpha_i,\ i = 1, \ldots, m$. On-boundary SVs are characterized by $0 < \alpha_i < C$ and $\xi_i = 0$, they constrain the width of the margin, namely, those lying on the hyperplanes $\mathbf{w}^T\mathbf{x} + b = \pm 1$. Off-boundary SVs have $\alpha_i = C$, $\xi_i > 0$, and non-support vectors are defined by $\alpha_i = 0$ and $\xi_i = 0$.

The standard SVM (2.1)-(2.2) has no feature selection instrument included, however it is still possible to build an SVM classifier that eliminates irrelevant features by using the ℓ_1-norm in the problem formulation. The next section explains the background and details of this approach.

2.3. Sparse SVMs

It was shown in [19], that for any point $\mathbf{q} \in \mathbb{R}^n$, not lying on the plane $P \doteq \{\mathbf{x} : \mathbf{w}^T\mathbf{x} + b = 0\}$, the distance between $\mathbf{q}$ and its projection on P, $p(\mathbf{q})$, is given by

$$\|\mathbf{q} - p(\mathbf{q})\| = \frac{|\mathbf{w}^T\mathbf{q} + b|}{\|\mathbf{w}\|'}, \tag{2.3}$$

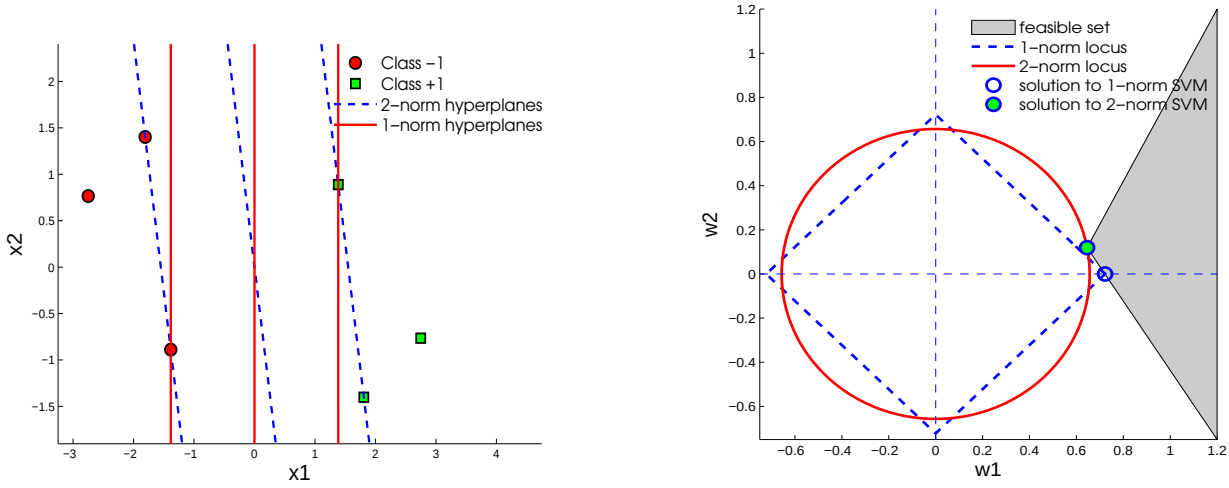
where $\|\cdot\|$ denotes a general norm, $\|\cdot\|'$ is the norm dual to $\|\cdot\|$, see the definition in Section 1.2. Based on this result, the following is true:

$$
\begin{aligned}
\|\mathbf{q} - p(\mathbf{q})\|_2 &= \frac{|\mathbf{w}^T\mathbf{q} + b|}{\|\mathbf{w}\|_2}, \\
\|\mathbf{q} - p(\mathbf{q})\|_1 &= \frac{|\mathbf{w}^T\mathbf{q} + b|}{\|\mathbf{w}\|_\infty}, \\
\|\mathbf{q} - p(\mathbf{q})\|_\infty &= \frac{|\mathbf{w}^T\mathbf{q} + b|}{\|\mathbf{w}\|_1}.
\end{aligned} \tag{2.4}
$$

Thus, if, e.g., the ℓ_2-norm is used to measure the distance between the planes $P_1 \doteq \{\mathbf{x} : \mathbf{w}^T\mathbf{x} + b = -1\}$ and $P_2 \doteq \{\mathbf{x} : \mathbf{w}^T\mathbf{x} + b = 1\}$, the margin (distance) between the planes P_1 and P_2 is $2/\|\mathbf{w}\|_2$, as we have mentioned before, see also [20]. Similarly, if the ℓ_∞-norm is used to measure the distance between the planes, the margin is $2/\|\mathbf{w}\|_1$, as the ℓ_∞-norm and ℓ_1-norm are dual. To maximize the margin $2/\|\mathbf{w}\|_1$, we minimize $\|\mathbf{w}\|_1$, which yields the following optimization problem that we call the linear *sparse* support vector machine (SSVM):

$$
\begin{aligned}
\underset{\mathbf{w},b,\boldsymbol{\xi}}{\text{minimize}} \quad & \|\mathbf{w}\|_1 + C\mathbf{e}^T\boldsymbol{\xi} \\
\text{subject to} \quad & \mathbf{D}(\mathbf{X}\mathbf{w} + b\mathbf{e}) \geq \mathbf{e} - \boldsymbol{\xi}, \\
& \boldsymbol{\xi} \geq \mathbf{0}.
\end{aligned} \tag{2.5}
$$

Figure 2.2 contains a two-dimensional example contrasting the geometry of the SSVM (2.5) and SVM (2.1) with $b = 0$ and $\mathbf{w} = (w_1, w_2)^T$ and illustrating how sparsity is induced by the ℓ_1-norm. The solution of the sparse SVM has the second component $w_2 = 0$ due to the pointed shape of the locus of points of $\|\mathbf{w}\|_1$; this geometry is the source of the sparsity. Note that problem (2.5) contains absolute values in the objective function: $\|\mathbf{w}\|_1 = \sum_{i=1}^n |w_i|$. To overcome this, we introduce non-negative variables $\mathbf{w}^+$ and $\mathbf{w}^-$ such that $\mathbf{w} = \mathbf{w}^+ - \mathbf{w}^-$,



(a) (b)

Figure 2.2: Two-dimensional toy data experiment: (a) ℓ_1-norm and ℓ_2-norm separating hyperplanes; (b) loci of points in the feature space and SVM solutions corresponding to ℓ_1-norm and ℓ_2-norm regularization.

and (2.5) can be converted to the following linear programming (LP) problem [21]:

$$\begin{aligned} \underset{\mathbf{w}^+,\mathbf{w}^-,b,\boldsymbol{\xi}}{\text{minimize}} \quad & \mathbf{e}^T(\mathbf{w}^+ + \mathbf{w}^-) + C\mathbf{e}^T\boldsymbol{\xi} \\ \text{subject to} \quad & \mathbf{D}(\mathbf{X}(\mathbf{w}^+ - \mathbf{w}^-) + b\mathbf{e}) \geq \mathbf{e} - \boldsymbol{\xi}, \\ & \mathbf{w}^+, \mathbf{w}^-, \boldsymbol{\xi} \geq \mathbf{0}. \end{aligned} \tag{2.6}$$

The dual of the problem (2.6) is also an LP:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & \mathbf{e}^T\boldsymbol{\alpha} \\ \text{subject to} \quad & -\mathbf{e} \leq \mathbf{X}^T\mathbf{D}\boldsymbol{\alpha} \leq \mathbf{e} \\ & \mathbf{e}^T\mathbf{D}\boldsymbol{\alpha} = 0, \\ & 0 \leq \boldsymbol{\alpha} \leq C\mathbf{e}. \end{aligned} \tag{2.7}$$

To find the optimal solution for LPs (2.6)-(2.7), we use the primal dual interior point method described in [22], see Section 2.4. This is a one-phase path-following method that can start from an infeasible point and lead directly to the optimal solution. An advantage of this approach is that one can monitor the variation of the primal and dual variables simultaneously.

By introducing additional nonnegative variables b^+ and b^- such that $b = b^+ - b^-$, we can convert the problem (2.6) into a LP of the form:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{2.8}$$

where $\mathbf{x} = (\mathbf{w}^+, \mathbf{w}^-, b^+, b^-, \boldsymbol{\xi})^T \in \mathbb{R}^{2n+m+2}$, $\mathbf{c} = (\underbrace{1, 1 \ldots, 1}_{2n}, 0, 0, \underbrace{C, C, \ldots, C}_{m})^T$, $\mathbf{b} = (\underbrace{1, 1, \ldots, 1}_{m})^T$, and the matrix $\mathbf{A} \in \mathbb{R}^{m\times(2n+m+2)}$ has the form: $\mathbf{A} = [\mathbf{DX}, -\mathbf{DX}, \mathbf{De}, -\mathbf{De}, \mathbf{I}_m]$. We consider how to solve LP (2.8) by the primal dual interior point method (PDIPM) in the next section.

2.4. Primal Dual Interior Point Method

To start, we introduce non-negative slack variables $\mathbf{u}$, and problem (2.8) is converted to the following LP:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} - \mathbf{u} = \mathbf{b}, \\ & \mathbf{x}, \mathbf{u} \geq \mathbf{0}. \end{aligned} \tag{2.9}$$

The inequality constraints are then replaced with extra terms in the objective function:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{c}^T\mathbf{x} + \mu \sum_i \log x_i + \mu \sum_j \log u_j \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} - \mathbf{u} = \mathbf{b}, \end{aligned} \tag{2.10}$$

where μ is a positive barrier parameter. As μ varies, the minimizers $(\mathbf{x}(\mu), \mathbf{u}(\mu))$ form the central path inside the feasible region, and as μ gets closer to zero, this sequence of solutions approaches the optimal solution of LP (2.9). The Lagrangian for our problem is $L(\mathbf{x}, \mathbf{u}, \mathbf{p}) = \mathbf{c}^T\mathbf{x} + \mu \sum_i \log x_i + \mu \sum_j \log u_j - \mathbf{p}^T(\mathbf{b} - \mathbf{A}\mathbf{x} + \mathbf{u})$, where $\mathbf{p}$ is the vector of dual variables. Taking derivatives of $L(x, u, p)$ with respect to each variable and setting them to zero, we get the Karush-Kuhn-Tucker (KKT) first-order optimality conditions [22]:

$$\begin{aligned} \mathbf{A}^T\mathbf{p} + \mu\mathbf{X}^{-1}\mathbf{e} &= \mathbf{0}, \\ \mathbf{p} - \mu\mathbf{U}^{-1}\mathbf{e} &= \mathbf{0}, \\ \mathbf{A}\mathbf{x} - \mathbf{u} &= \mathbf{b}, \end{aligned} \tag{2.11}$$

where $\mathbf{X}$ and $\mathbf{U}$ are diagonal matrices with the components of $\mathbf{x}$ and $\mathbf{u}$ on the diagonals, respectively. Introducing $\mathbf{z} = \mu\mathbf{X}^{-1}\mathbf{e}$, equations (2.11) can be written in the form:

$$\begin{aligned} \mathbf{A}^T\mathbf{p} + \mathbf{z} &= \mathbf{c}, \\ \mathbf{A}\mathbf{x} - \mathbf{u} &= \mathbf{b}, \\ \mathbf{Z}\mathbf{X}\mathbf{e} &= \mu\mathbf{e}, \\ \mathbf{P}\mathbf{U}\mathbf{e} &= \mu\mathbf{e}, \end{aligned} \tag{2.12}$$

where $\mathbf{P}$ and $\mathbf{Z}$ are diagonal matrices of $\mathbf{p}$ and $\mathbf{z}$, respectively. Note that the first two equations in (2.12) are primal and dual constraints, respectively, and the last two equations imply complementary slackness[2].

The idea of PDIPM is to solve the system of equations (2.12) using Newton's method. Starting with an initial positive values $\mathbf{x}$, $\mathbf{u}$, $\mathbf{z}$, and $\mathbf{p}$, our aim is to find a step direction $(\Delta\mathbf{x}, \Delta\mathbf{u}, \Delta\mathbf{z}, \Delta\mathbf{p})$ such that the new point $(\mathbf{x}+\Delta\mathbf{x}, \mathbf{u}+\Delta\mathbf{u}, \mathbf{z}+\Delta\mathbf{z}, \mathbf{p}+\Delta\mathbf{p})$ lies approximately on the primal-dual central path at the point $(\mathbf{x}(\mu), \mathbf{u}(\mu), \mathbf{z}(\mu), \mathbf{p}(\mu))$. If so, it should satisfy equations (2.12). Plugging the point $(\mathbf{x}+\Delta\mathbf{x}, \mathbf{u}+\Delta\mathbf{u}, \mathbf{z}+\Delta\mathbf{z}, \mathbf{p}+\Delta\mathbf{p})$ into equations(2.12), then simplifying and dropping non-linear terms, we obtain:

$$\begin{aligned}
&\mathbf{A}^T\Delta\mathbf{p} + \Delta\mathbf{z} = \mathbf{c} - \mathbf{A}^\mathbf{T}\mathbf{p} - \mathbf{z} := \boldsymbol{\rho},\\
&\mathbf{A}\Delta\mathbf{x} - \Delta\mathbf{u} = \mathbf{b} - \mathbf{A}\mathbf{x} + \mathbf{u} := \boldsymbol{\sigma},\\
&\mathbf{Z}\Delta\mathbf{x} + \mathbf{X}\Delta\mathbf{z} = \mu\mathbf{e} - \mathbf{X}\mathbf{Z}\mathbf{e},\\
&\mathbf{U}\Delta\mathbf{p} + \mathbf{P}\Delta\mathbf{u} = \mu\mathbf{e} - \mathbf{P}\mathbf{U}\mathbf{e}.
\end{aligned} \tag{2.13}$$

Note that the system of equations (2.13) can be reduced further as the last two equations are trivial and can be eliminated by solving them for $\Delta\mathbf{z}$ and $\Delta\mathbf{u}$, and then substituting the results into the first two equations. We get the *reduced KKT* system:

$$\begin{aligned}
&\mathbf{A}^T\Delta\mathbf{p} - \mathbf{X}^{-1}\mathbf{Z}\Delta\mathbf{x} = \boldsymbol{\rho} - \mu\mathbf{X}^{-1}\mathbf{e} + \mathbf{z}\\
&\mathbf{A}\Delta\mathbf{x} - \mathbf{P}^{-1}\mathbf{U}\Delta\mathbf{p} = \boldsymbol{\sigma} + \mu\mathbf{P}^{-1}\mathbf{e} - \mathbf{u}.
\end{aligned} \tag{2.14}$$

Before summarizing the algorithm, we need to know how to compute μ and determine the step length parameter θ. Complementarity measure μ is defined by $\mu = \delta\frac{\gamma}{l+k}$, where

[2]That is, for the optimal solutions to the primal and the dual, for any variable that is set to a positive value in the primal (dual), the corresponding slack variable in the dual (primal) must be set to zero. Conversely, if all of these constraints are satisfied for a pair of feasible solutions, then these solutions must be optimal.

$\gamma = \mathbf{z}^T\mathbf{x} + \mathbf{p}^T\mathbf{u}$, $0 \leq \delta \leq 1$, and l and k are the lengths of $\mathbf{x}$ and $\mathbf{p}$, respectively. A small value of γ translates into a small duality gap $|\mathbf{c}^T\mathbf{x} - \mathbf{b}^T\mathbf{p}|$. To keep variables positive, the step length parameter θ is chosen as minimum out of $0.9/(\max(\frac{-\Delta\mathbf{x}}{\mathbf{x}}, \frac{-\Delta\mathbf{u}}{\mathbf{u}}, \frac{-\Delta\mathbf{z}}{\mathbf{z}}, \frac{-\Delta\mathbf{p}}{\mathbf{p}}))$ and 1. For a stopping rule we take $\max\{\gamma, \|\boldsymbol{\rho}\|_1, \|\boldsymbol{\sigma}\|_1\} \leq \epsilon$ for a given tolerance ϵ, provided that values $\|\mathbf{x}\|_\infty$, $\|\mathbf{p}\|_\infty$, and $|b|$ are no too large [22]. The method is summarized in Algorithm 1.

Algorithm 1: PDIPM Algorithm (Reduced KKT)

1 Initialize $(\mathbf{x}, \mathbf{u}, \mathbf{z}, \mathbf{p}) > \mathbf{0}$
2 **While** $\max\{\gamma, \|\boldsymbol{\rho}\|_1, \|\boldsymbol{\sigma}\|_1\} > \epsilon$ **repeat** {
3 Compute $\boldsymbol{\rho}, \boldsymbol{\sigma}, \gamma, \mu$
4 Solve system of equations (2.14)
5 Determine the step length θ
6 Set $(\mathbf{x}, \mathbf{u}, \mathbf{z}, \mathbf{p}) := (\mathbf{x}, \mathbf{u}, \mathbf{z}, \mathbf{p}) + \theta(\Delta\mathbf{x}, \Delta\mathbf{u}, \Delta\mathbf{z}, \Delta\mathbf{p})$ }

2.5. Summary

In this chapter, we introduced linear sparse SVMs, solved by the primal dual interior point method, as a tool for simultaneous classification and feature selection. The examples of the SSVMs usage are given in Chapter 3 (hyperspectral band selection) and Chapter 4 (classification on embedded Grassmannians).

CHAPTER 3

Hyperspectral Band Selection Using Sparse Support Vector Machines

3.1. Introduction

A digital hyperspectral image can be considered as a three-dimensional array consisting of two spatial dimensions and one spectral dimension. The spectral dimension consists of images collected across tens to hundreds narrow wavelength bands and combined to form a hyperspectral data cube, see Figure 3.1. Thus, each pixel in the data cube acquires many bands of light intensity data from the spectrum, extending the RGB (red, green, and blue) color model beyond the visible. Hyperspectral imaging (HSI) is used in various applications, e.g., material identification, land cover classification, or surveillance [1].

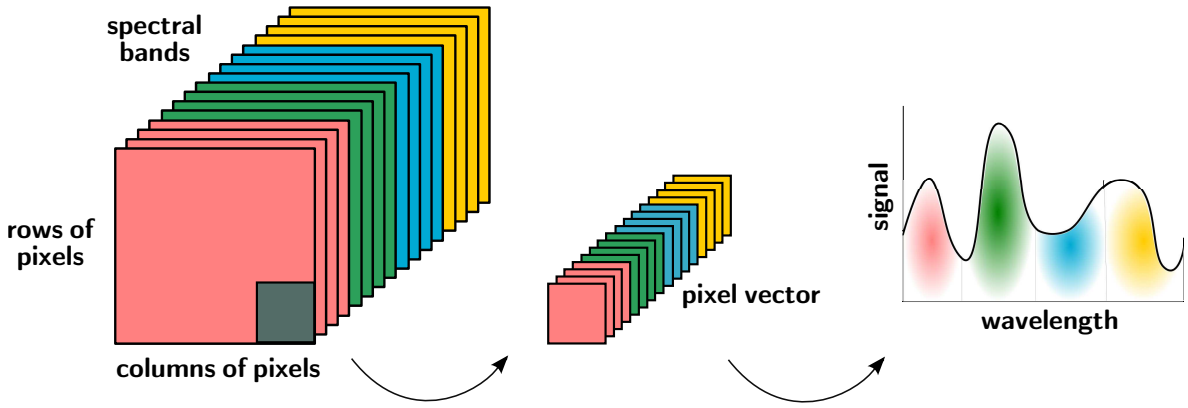


Figure 3.1: Hyperspectral data.

It is now well established that HSI contains an abundance of useful information beyond the visible spectrum [1]. However, processing snapshots of high-dimensional hyperspectral data has proven to be a formidable computational and algorithmic challenge. Information amongst the bands may be highly correlated suggesting that appropriate subset selection

could be beneficial. It has also been observed that more bands are not necessarily better and adding bands can actually degrade algorithm performance, a phenomenon referred to as the Hughes effect [23]. Thus, a pre-processing step is often necessary to reduce the data volume and remove information redundancy for subsequent data analysis, and it can be realized by using band selection techniques. In band selection, the goal is to identify a subset of bands in the spectrum that contain the most discriminatory information during a particular classification task, without losing the classification accuracy. Band selection is of particular interest in building models for specific applications such as the detection and discrimination of chemical plumes where signatures of known chemical vapors are available.

Three general approaches to hyperspectral band selection problem have been proposed: filters, wrappers and embedded methods [4]. A filtering method example is the band add-on (BAO) algorithm in which selected bands increase the spectral angle mapper measure between two spectra [24]. Filtering algorithms presented in [25] and [26] (or [27]) are based on mutual information (MI), an absolute measure of independence or common information between two random sources. Wrappers perform feature selection for a specific classifier using its accuracy to evaluate the importance of each feature [28]. In [29], a wrapper-based genetic algorithm (GA) was combined with a SVM [5] for hyperspectral feature selection. Wrapper methods treat the selected classifier as a black box, i.e. feature selection does not depend on its internal mechanism. In contrast to filters and wrappers, embedded methods are specific to the chosen learning machine as they select features as part of the process of training. There are different embedded approaches, including forward-backward methods, optimization of scaling factors, and use of a sparsity term in the objective function [30]. SVM Recursive Feature Elimination (SVM-RFE), proposed in [31], uses the SVM feature weight magnitudes as ranking criterion during a greedy backward selection process. In

[32], SVM-RFE is compared to EFS-SVM, Embedded Feature Selection SVM algorithm for hyperspectral images. The EFS-SVM embeds a weighting into the SVM kernel function and iteratively updates the weights using a logistic function measuring each band importance. The Recursive SVM (R-SVM) and its modification, MR-SVM, in [33] train an SVM and calculate discriminatory power for each band from its weight in a backward elimination procedure.

Recent trends in data analysis have seen a rise in popularity of sparsity inducing penalty functions, in particular, the ℓ_1-norm penalty. This approach is attractive given ℓ_1-norm optimization problems are readily handled via fast convex solvers and serve as a proxy for ℓ_0-norm optimization problems which are prohibitively expensive. The ℓ_1-norm penalty was initially proposed in the context of linear SVMs in [17], and also used in [16], [18] and [34]. This methodology was used for dimensionality reduction in the context of drug design, based on training linear support vector regression (SVR) models for selecting features and then creating a nonlinear SVR model for reduced data classification [35]. The authors also used the bootstrap aggregating approach of [36].

An improved hybrid ℓ_1-norm SVM to reduce noise features was proposed in [37]. The geometry of the SVM with the ℓ_1-norm regularization results in feature weights being set to zero effectively, i.e., they serve as embedded feature selectors.

To our knowledge, the sparsity inducing ℓ_1-norm SVMs, or sparse SVMs, described in Chapter 2, has not been exploited in the context of hyperspectral embedded band selection. We will develop a new band selection procedure whose characteristics can be summarized as follows:

- A linear SSVM is used as a basic model for band selection. Unlike [17], [18], or [34], it is solved by the primal dual interior point method (Section 2.4) that allows one to monitor the variation of the primal and dual variables simultaneously.
- We exploit the nature of the sparsity of the SSVM algorithm and propose a weight ratio criterion for embedded band selection. Unlike other variations of SVM, this approach, when used with SSVMs, easily distinguishes the non-zero weights from the zero weights in an objective manner, a feature that is critical to the implementation of the band selection problem. The usual SVM method of selecting features from the weights with the largest magnitudes fails to provide a rational means for band selection in hyperspectral imagery.
- Motivated by [35], we employ the bootstrap aggregating approach of [36] to enhance the robustness of sparse support vector machines. In contrast to [35], we restrict our attention to linear SSVMs so that we only need to tune one learning parameter.
- We extend the binary band selection to the multiclass case by proposing three approaches combined with one-against-one (OAO) SSVMs. Two of them are extensions of the SSVM Algorithm based on pairwise band selection between classes. The third proposed method is a combination of the filter band selection method WaLuMI [27] in sequence with the OAO SSVM which serves to reduce more bands via the embedded feature selection properties of the algorithm.
- We apply the SSVM algorithm to the HSI classification problem, and show that it is an effective technique for embedded band selection while at the same time achieving competitively high accuracies in benchmark numerical experiments.

This chapter is organized as follows. Section 3.2 covers the SSVM band selection framework. The experimental results are presented in Section 3.3, followed by conclusion remarks in Section 3.4.

3.2. Band Selection via SSVMs

In this section we describe our sparse SVM approach to the hyperspectral band selection problem. We consider the band selection algorithm for two-class data problem as well as its extension to a multiclass data using bands selected from pairwise modeling approach.

3.2.1. Band Selection: Binary Case. We now describe the two-class band selection algorithm. The sparsity of the SSVM weight vector $\mathbf{w}$ identifies bands that are candidates for elimination. Given the data is inherently noisy there is a stochastic variability in the vectors $\mathbf{w}$ and in the bands selected. In a fashion similar to [35], we address this variability using **b**ootstrap **agg**regat**ing** (bagging) [36]. In [35], the authors used bagging to reduce variability and obtain bagged SV regression (SVR) variable selection and nonlinear SVR classification models. We adopt the bagging technique to train our SSVM to make our selection model more robust. We replicate the training data set N times by sampling randomly with replacement. For each pair of classes, N SSVM models are generated based on these N sets, each resulting in a different weight vector $\mathbf{w}$. As a result, for each band there is a set (or a sample) of N weight values taken from different $\mathbf{w}$'s. To reduce the number of bands, we eliminate those with at least 95% of "zeros" in the samples.

We illustrate the impact of the ℓ_1-norm on the solution in Figure 3.2. Both ℓ_1-norm and ℓ_2-norm SVMs are trained on two classes from the AVIRIS Indian Pines data set [38], described in Section 3.3.2. In contrast to the standard SVM, that uses all the bands for

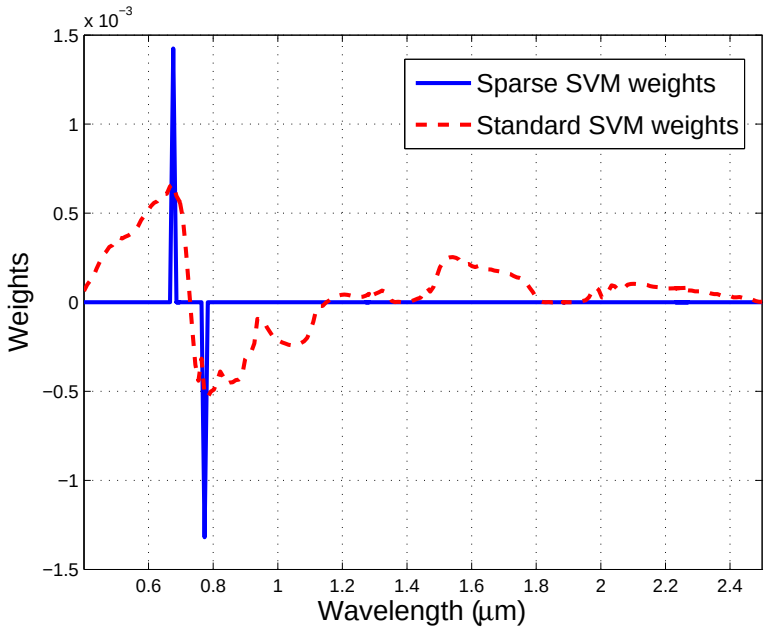


Figure 3.2: Comparison of weights for sparse SVM and standard SVM models using class Corn-min and class Woods of the AVIRIS Indian Pines Data Set. Note that SSVM selects two non-zero weights while standard SVM has a weight profile that matches the differential in signature between the two classes.

discrimination, the sparse SVM identifies two bands (out of a total of 220) that can be used to separate the two classes.

After the bagging step, an SSVM is trained on the reduced data, and the resulting SSVM weights are ordered by magnitude. Comparing magnitude orders, we can eliminate more bands: if $\frac{|w_i|}{|w_{i+1}|} = O(10^M)$ and $M > 1$ for some i^*, we remove bands starting from index $i^* + 1$. For instance, in our experiments in Section 3.3.2, we have observed that $M = 5$, i.e., there is a sharp transition separating the zero from non-zero weights. We provide numerical results in Table 3.2 to support this observation.

The method is summarized in Algorithm 2.

Algorithm 2: Two-class Band Selection SSVM Algorithm

Input: Training data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, class labels $d_i \in \{-1, +1\}$, $i = 1, \ldots, m$, set of kept bands $S = \{1, 2, \ldots, n\}$
Step 1. Variability Reduction.
Sample with replacement from $\mathbf{X}$ to obtain replicate training sets $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$
Train N SSVM models $f_j(\mathbf{x}) = (\mathbf{w}^j)^T\mathbf{x} + b^j \rightarrow$ weight vectors $\mathbf{w}^j$, $j = 1, \ldots, N$
For $k = 1 : n$, remove kth band if $\#\{|w_k^j| < \text{tolerance}, j = 1, \ldots, N\} \geq 0.95 * N$, $\rightarrow$ update S
Restrict $\mathbf{X}$ to selected bands: $\mathbf{X}_{new} = \mathbf{X}(:, S)$
Step 2. Final Selection.
Train an SSVM model f on $\mathbf{X}_{new} \rightarrow \mathbf{w}$
Rank $\mathbf{w}$ values by magnitude $\rightarrow \mathbf{w}^r$, keep ranked band indices in R
Go through $\mathbf{w}^r$ and compare magnitude orders: if $|w_{i_k}^r|/|w_{i_{k+1}}^r| = O(10^M)$ and $M > 1$ for some $k = k^*$, remove bands from R starting from index $i_{k*+1} \rightarrow$ update $S = S \setminus S(R)$
Restrict $\mathbf{X}_{new}$ to selected bands: $\mathbf{X}_{new} = \mathbf{X}_{new}(:, S)$
Output: Band selected list S, linear SSVM model f

3.2.2. Multiclass Band Selection. Hyperspectral images typically consist of more than two classes of data, therefore we consider possible extensions of the binary Algorithm 2 to the multiclass case by proposing three methods.

Methods I and II concern using the set of bands selected in the context of binary models. This allows us to use the results of the embedded band selection described above. Hence, after performing binary band selection for all pairs of c classes, we have $\binom{c}{2} = c(c-1)/2$ subsets of selected bands. Note that simply taking the superset or intersection of these subsets is not an option in general as the superset can be equal to the original set of bands, and the intersection can be the empty set. Our third approach differs from the two above in that it is a combination of a filter method and OAO SSVMs.

- **Method I:** Rank selected bands by the frequency of their occurrence in all the two-class subsets and select K bands with the highest frequency values for a chosen number K.

- **Method II:** Rank bands in each two-class subset by magnitude and take the superset of the T top bands from each subset. For simplicity, $T = 1$ is taken, in which case the method gives only a fixed set of selected bands.
- **Method III:** This approach does not use the results of the two class band selection problem. The well-known Ward's Linkage Strategy Using Mutual Information (WaLuMI) method [27] is employed as a pre-selection filter technique, briefly discussed in Section 3.3.2. This filter band selection step is followed by an application of the OAO SSVM which implicitly performs an embedded band selection in view of the sparse penalty term which effectively sets redundant weights to zero.

As mentioned in Section 3.1, for all the three methods, we adopt one-against-one (OAO) multiclass approach [1] to compare our results with other methods in the literature. It is based on defining a combined decision function on a set of binary classifiers. Given c classes, we build all pairwise $\binom{c}{2}$ ℓ_1-norm binary classifiers f_{ij}, taking training points from classes i and j, respectively. For a testing pixel $\mathbf{x}$, if f_{ij} determines the class of $\mathbf{x}$ to be i, we increase the vote for class i by one. Otherwise, the vote for class j is increased by one. We repeat this for all classifiers, and the class with the largest number of votes is assigned to $\mathbf{x}$.

3.3. Experimental Results

Now we present computational results both for binary and multiclass band selection and classification and compare them with other techniques.

3.3.1. Comparison With Other Methods. The computational results include performance of the method on the AVIRIS Indian Pines [38] and Long-Wavelength Infrared (LWIR) [40] data sets. We apply the SSVM algorithm to the binary classification problem

[1]We note that the classification results obtained using one-against-all (OAA) SVMs [39] were inferior compared to those obtained using OAO SVMs.

on both data bases, and also compare the results of the multiclass SSVM algorithm on the Indian Pines data set with several other well-known techniques found in the literature. Results for both the two class problem and multiclass problem are analyzed. The techniques used for comparison are briefly summarized below:

(1) *WaLuMI:* Ward's Linkage strategy Using Mutual Information (WaLuMI) [26] (or [27]) is a filtering band selection technique that uses no supervised information. It is a hierarchical clustering approach that exploits band correlation using a mutual information (MI) criterion. According to WaLuMI, bands are grouped "to minimize the intracluster variance and maximize the intercluster variance" [27]. A distance matrix used in a clustering process is calculated using MI. A final set of bands is selected as a set of representative bands from each group such that each selected band has the highest average correlation (mutual information) with regard to the other bands in the corresponding cluster. After the band selection process is done, any classification method can be performed on the reduced data to obtain classification accuracy rates. We compare the results of this method to our binary and multiclass SSVM results. In addition to comparing the results from WaLuMI as described in [26, 27] to our results, we propose its application in a preprocessing state of the $c > 2$ class problem. For implementation of WaLuMI for Method III we used the software *BandSelection_TGRS07* [41].

(2) *B-SPICE:* Proposed in [42], this method performs simultaneous band selection and endmember detection. It extends the SPICE, the Sparsity Promoting Iterated Constrained Endmember algorithm, with integrated band selection. It is done by adding band weights and a band sparsity promoting term (BST) to the SPICE objective

function. The method is a filter, and after selecting the relevant bands, the authors performed one-against-one Relevance Vector Machine (RVM) classification. We used this method for comparing results for the multiclass data.

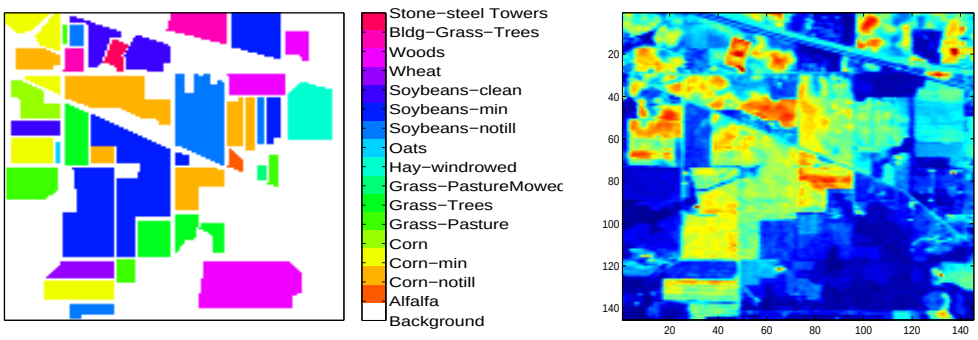
(3) *Lasso Logistic Regression:* The Lasso logistic regression, or ℓ_1-norm regularized logistic regression, proposed in [6], has become a popular tool for data classification. We solve the following optimization problem:

$$\min_{\beta_0,\boldsymbol{\beta}} -\frac{1}{m}\sum_{i=1}^{m} y_i(\beta_0 + \mathbf{x}_i^T\boldsymbol{\beta}) - \log(1 + e^{(\beta_0+\mathbf{x}_i^T\boldsymbol{\beta})}) + \lambda\|\boldsymbol{\beta}\|_1,$$

where $(\beta_0, \boldsymbol{\beta})$ are the model parameters, λ is a tuning parameter, m is the number of data points $\mathbf{x}_i$, and y_i are response variables. The ℓ_1-norm induces sparsity in the parameter, with zero components corresponding to redundant bands. We implement this approach for binary band selection only, via available R-based glmnet-package [43].

3.3.2. AVIRIS Indian Pines Data Set. The hyperspectral Indian Pines data set was collected by an Aiborne Visible/Infrared Imaging Spectrometer (AVIRIS) over a small agricultural area in Northwestern Indiana in 1992 [38]. It consists of 145×145 pixels by 220 bands from 0.4 to 2.4 μm.[2] Note, that in the literature, water absorption bands $104 - 108$, $150 - 163$, and 220 are often discarded before experiments. In our experiments we include all the 220 original bands with the idea that the band selection algorithm should ignore these bands if it is performing as we expect. Figure 3.3 shows the image at band 31 ($\sim 0.7\mu$m) and the ground truth of the scene. Due to availability of the ground truth, 10366 pixels were prelabeled to be in one of the 16 classes. The unlabeled background pixels are not used in

[2]We note that it is common practice in the literature for this data set to refer to bands using their indices, rather than the wavelength values, and we follow that convention here.



(a) (b)

Figure 3.3: AVIRIS Indian Pines data set: (a) ground truth; (b) one band image.

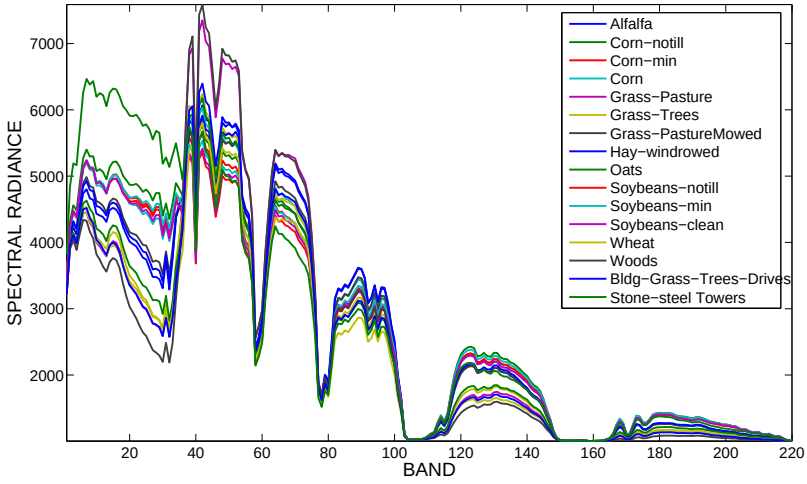


Figure 3.4: Averaged spectral signatures of the Indian Pines data set classes.

our experiments. Figure 3.4 depicts averaged spectral radiance curves for each class, with the radiance units being watts $* \, \mathrm{cm}^{-2} * \mathrm{nm}^{-1} * \mathrm{sr}^{-1}$. We preprocessed the data by finding the mean over all the pixels and then subtracting it from each pixel in the scene.

The data was randomly partitioned into 50% for training and 50% for testing (Table 3.1). The training set was used to build SSVM models using bootstrap aggregating [36]. The values of penalty parameter C were found by performing 5-fold cross-validation on the training data. The number N of data bootstrap samples used in the SSVM Algorithm was set to 100.

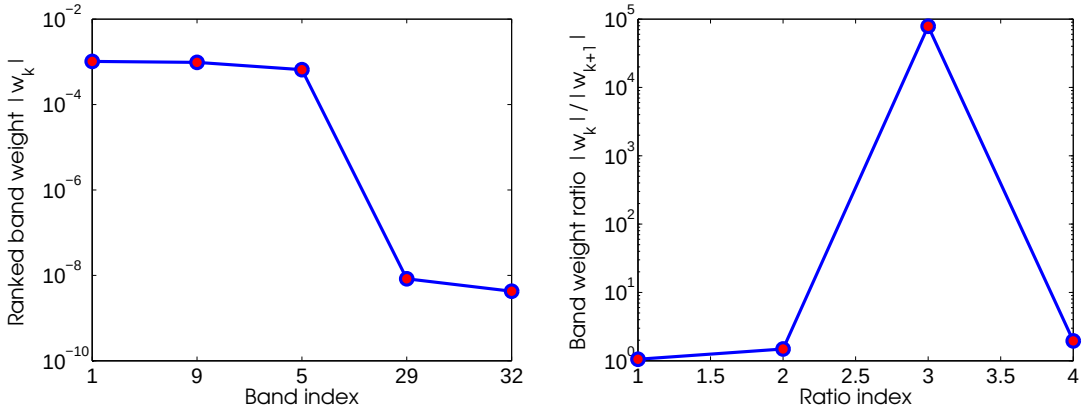
Table 3.1: The Indian Pines data set: number of training and testing pixels in each class.

Class	**# Training Points**	**# Testing Points**
Alfalfa	27	27
Corn-notill	717	717
Corn-min	417	417
Corn	117	117
Grass/Pasture	249	248
Grass/Trees	374	373
Grass/Pasture-mowed	13	13
Hay-windrowed	245	244
Oats	10	10
Soybeans-notill	484	484
Soybeans-min	1234	1234
Soybeans-clean	307	307
Wheat	106	106
Woods	647	647
Bldg-grass-trees-drives	190	190
Stone-steel towers	48	47
Total	**5185**	**5181**

Table 3.2: The magnitude of ordered weights obtained using the SSVM Algorithm. SSVM produces a steep drop in the weight values. Only bands associated with the non-zero weights are selected, i.e., before the steep drop in their magnitude.

Corn-min and Woods		**Corn-notill and Grass/Trees**	
Band	Weight	Band	Weight
29	1.4249e-03	1	1.0202e-03
41	1.3191e-03	9	9.6991e-04
28	3.5594e-08	5	6.5283e-04
42	1.6342e-09	29	8.3022e-09
27	1.3258e-09	32	4.2466e-09
...	...	...	...

Using the experimental setup described above, we apply our two-class band selection SSVM Algorithm to the Indian Pines data set. Table 3.2 lists several top weights ordered by magnitude at the final selection step of Algorithm 2. The distinction between the zero and non-zero weights is made clearly by the large gap $O(10^5)$ in the magnitudes determined by the ratios. For two pairs of classes, Corn-min and Woods, and Corn-notill and Grass/Trees,



(a) (b)

Figure 3.5: SSVM band selection for Corn-notill and Grass/Trees given the subset of bands (1,9,5,29,32) ranked by magnitude: (a) band weights $|w_k|$ vs. band indices; (b) band weight ratios $|w_k|/|w_{k+1}|$ vs. ratio indices. See also Table 3.2.

the sets of bands selected are (29,41) and (1,9,5), respectively. Figure 3.5 visualizes five top band weights $|w_k|$ and four corresponding ratios $|w_k|/|w_{k+1}|$ for classes Corn-notill and Grass/Trees according to Table 3.2. It is seen that the third ratio, corresponding to the ratio with original indices $|w_5|/|w_{29}|$, is of order $O(10^5)$, which suggests the removal of bands 29 and 32.

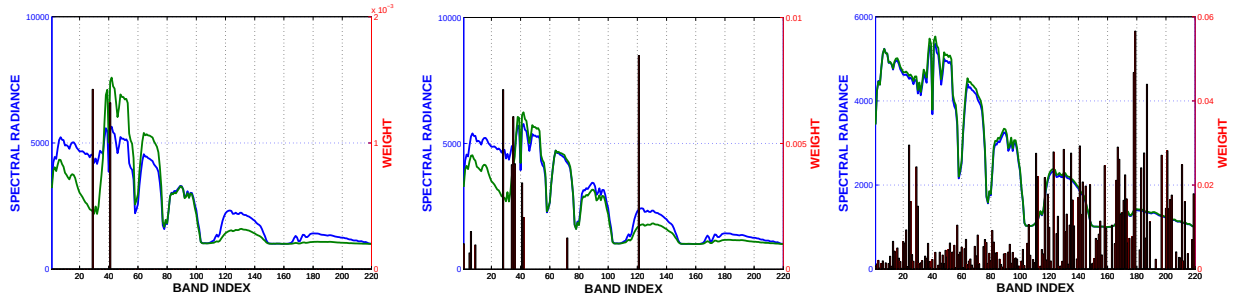
Table 3.3 shows the number of selected bands and classification accuracy for three pairs of classes in comparison to the other methods. These classes were selected to illustrate the diversity of performance of the method that is inherently dependent on the complexity and similarity of the signatures of interest. The bands that were selected for each pair of classes are shown in Figure 3.6 along with the spectral signatures. We plotted the difference between two spectral signatures and the corresponding band weights in Figure 3.7.

As an embedded method, the SSVM Algorithm selects bands that contribute most to the process of separating the classes. It is interesting to note that the SSVM selection for Corn-min and Woods pick only two bands, 29 and 41. These bands are located precisely

Table 3.3: Accuracy rates (%) for binary band selection.

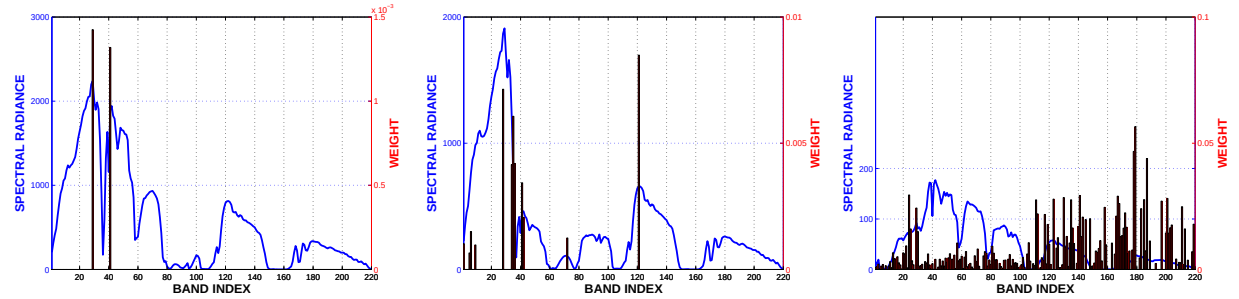
Classes	**Accuracy: all bands**	**SSVM Algorithm**		**WaLuMI + SSVM**		**Lasso Logistic Regression**	
		# Bands Kept	Accuracy	# Bands Kept	Accuracy	# Bands Kept	Accuracy
Corn-min and Woods	100.00	2	100.00	2	99.9	12	100.00
Corn-notill and Grass/Trees	99.73	12	99.73	12	100	19	98.9
Soybeans-notill and Soybeans-min	89.58	179	89.23	-	-	127	89.52

where the difference in the spectral signatures is the largest. When we run the WaLuMI algorithm with the number of bands preselected to be two we obtain bands 54 and 184. Both bands occur where the difference in the signatures is smaller than for the bands selected by SSVM. For the pair Corn-notill versus Grass/Trees the SSVM algorithm identified 12 bands: 121,28,35,36,34,41,42,6,72,1,9,5 (ranked by magnitude), while when WaLuMI is preselected to compute 12 bands, it identifies - 12,22,36,50,68,88,100,127,162,165,183,209. We note a tendency by WaLuMI to select high band indices while SSVM favors low indices. We note that the SSVM algorithm has identified neighboring spectra as being important in the model, e.g., bands (34, 35, 36), (5, 6) and (41, 42). One might infer that SSVM is characterizing these frequencies as very significant for inclusion. When we look at the plot of the difference in spectral signature, we observe that the difference in spectral signature is changing rapidly at these locations. One can speculate that the steepness of this curve requires more samples to capture accurately. We observe that for very similar classes, more bands are required to separate the data, as in case of Soybeans-notill and Soybeans-min, see Figure 3.7c. Apparently, the signatures are so similar that many more bands are required to discriminate between them. It is interesting to observe that for this case the Lasso logistic regression approach selected only 127 bands and demonstrated comparable accuracy. In



(a) (b) (c)

Figure 3.6: Spectral signatures and weights of selected bands for: (a) Corn-min and Woods, (b) Corn-notill and Grass/Trees, (c) Soybeans-notill and Soybeans-min.

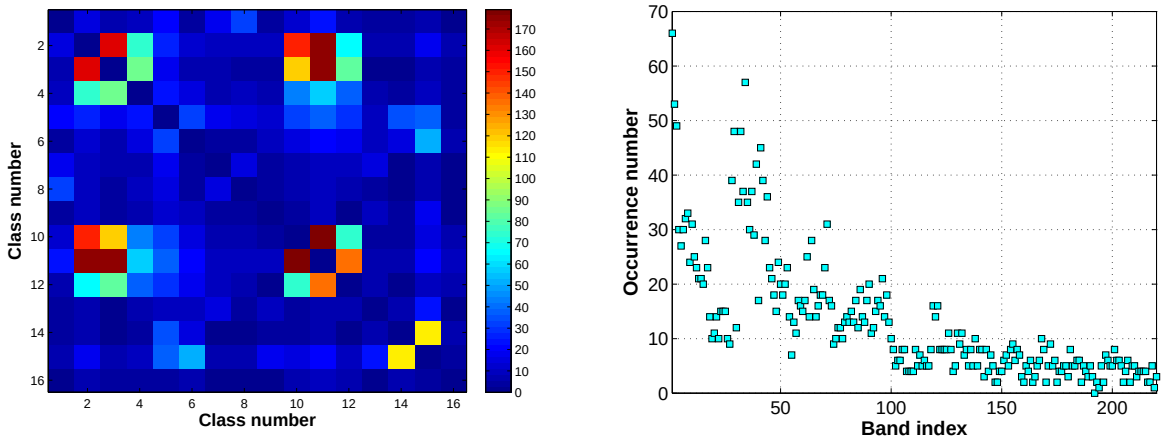


(a) (b) (c)

Figure 3.7: Difference plots of spectral signatures and weights of selected bands for: (a) Corn-min and Woods, (b) Corn-notill and Grass/Trees, (c) Soybeans-notill and Soybeans-min.

contrast, the Lasso logistic regression selected substantially more bands for the other cases with comparable classification rates.

After computing the discriminatory bands for all $\binom{16}{2} = 120$ pairs of Indian Pines classes according to the SSVM Algorithm, we implement the multiclass band selection described in Section 3.2.2. Figure 3.8a shows the distribution of number of bands selected for each pair of classes. It is apparent that some classes are so similar that sparse solutions do not exist, as in the case studied above for Soybeans-notill and Soybeans-min. Differences in class signatures are exploited by SSVM to identify optimal bands for classification, i.e., where the signatures



(a) (b)

Figure 3.8: Binary band selection for Indian Pines data: (a) a colormap reflecting the numbers of bands selected for each of 120 subsets, i.e., pairs of classes; (b) number of occurrences of each band.

are most distinct on average. When the spectral signatures are very similar there are no highly discriminatory bands for SSVM to select, and the net result is that the method needs to selects a large number of bands for successful discrimination. When the signatures are distinct, such as for Corn-min and Woods, substantially sparser models are able to model the decision function. According to Method I, the bands selected in the pairwise problems are ranked by their frequencies of inclusion, i.e., the number of times they were given non-zero values in the training phase. In Figure 3.8b the frequencies are given summed over all 120 pairs of classes. The bands with smaller indices appear to be more important in the multiclass problem.

The overall classification accuracy rates of Methods I, II, III for different subsets of selected bands are given in Table 3.4. The rates were obtained by training and testing multiclass OAO SSVMs on selected band sets and then performing *spatial smoothing* following [42]. Namely, for each test pixel, we consider its three by three contiguous neighborhood and assign the most frequently occurring class name in this neighborhood to the pixel. In

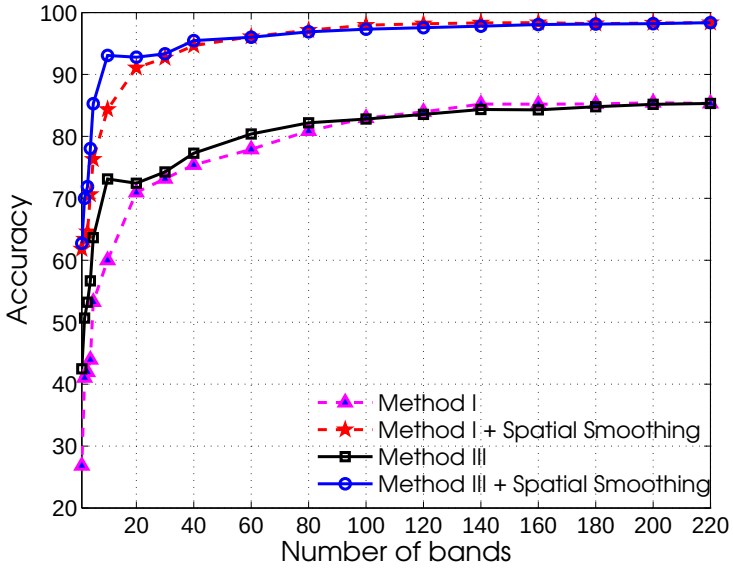


Figure 3.9: Accuracy plots for OAO SSVM before and after spatial smoothing obtained by Methods I and III.

this process we use the training pixels labels corresponding to the ground truth. Note that spatial smoothing improves classification rates significantly, see Figure 3.9.

Table 3.4 reveals interesting aspects of the relative performance of the algorithms. First we note that the combination of WaLuMI with SSVM (Method III) does not appear in prior literature. The SSVM will ignore bands selected by WaLuMI that it finds redundant, i.e., it will perform a secondary embedded band selection. We conclude that Method I is superior to Method III when we are including more bands but Method III outperforms Method I for smaller sets of bands. Both methods show a substantial improvement over other methods in the literature for the multiclass problem.

Method II (Table 3.4), as described in Section 3.2.2, gave a fixed set of 57 selected bands, with OAO SSVM with spatial smoothing classification accuracy on reduced data equal to 97.3%. This result is better than the corresponding results for Method I and Method III.

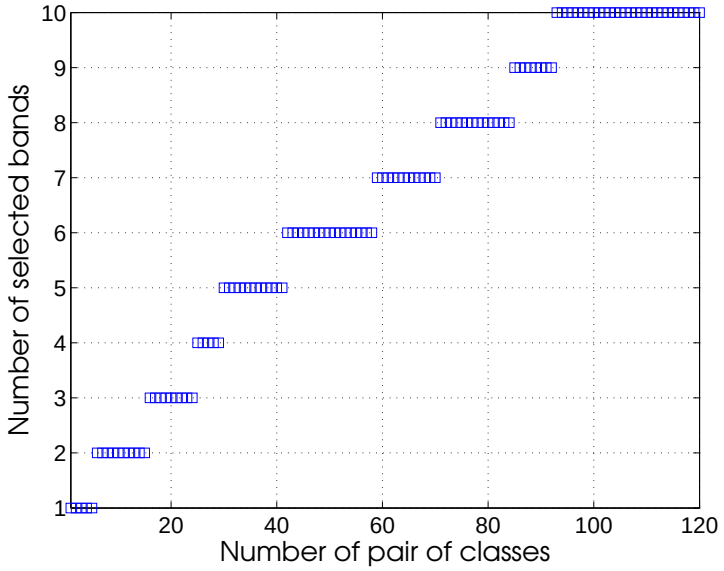


Figure 3.10: Number of bands selected by SSVM out of 10 bands preselected by WaLuMI for each pair of classes of the Indian Pines data set.

Note that among bands selected by Method II, there are no water absorption bands $104-108$, $150-163$, and 220. As for Method I, these noisy bands were selected in 5% pairs of classes.

Method III, as a combination of WaLuMI and SSVM, can be used for further data reduction. As we observe, the SSVM classifier drives to zero weights of some pre-selected by WaLuMI bands. Consider, for instance, the subset of ten bands selected by WaLuMI with indices $5, 25, 52, 55, 68, 79, 88, 100, 129, 183$. The OAO SSVM applied to the data with this set of bands, remove more bands for most pairs of classes. Figure 3.10 reflects the statistics: we can see how many bands are selected out of 10 for each of 120 pairs of classes. The results are sorted by number of bands.

We compared our results to those reported in [42] and [26]. We did not make comparisons with the WaLuMI experiments described in [27], as we did not use background pixels in our experiments. For comparison with [42], we used the results from the Table III in the paper, run 3 (see the B-SPICE + RVM column). For comparison with [26], we took the results from

Table 3.4: Accuracy results for multiclass band selection (%) and comparison with other methods.

# Bands Kept	Method I	Method II	Method III (WaLuMI + SSVM)	Comparison	
				B-SPICE + RVM [42]	WaLuMI + NN [26]
220	98.36	-	98.36	93.9	-
124	98.24	-	97.59	93.7	-
122	98.13	-	97.59	93.2	-
103	97.74	-	97.49	93.5	-
89	97.36	-	97.47	93.6	-
80	97.14	-	96.89	-	-
60	96.12	-	96.02	-	-
57	95.66	97.3	96.22	-	-
40	94.65	-	95.46	-	80
34	93.15	-	93.03	86.4	80
30	92.67	-	93.34	-	79
20	91.08	-	92.78	-	79
19	91.20	-	92.57	82.5	81
18	88.59	-	92.78	78.3	82
10	84.37	-	93.07	-	81
5	76.32	-	85.29	-	71

the paper, namely Nearest Neighbor (NN) classifier rates obtained on subsets determined by WaLuMI (the WaLuMI + NN column). Note that nearest neighbor classification becomes computationally prohibitive as the size of the image grows.

Table 3.5 shows the bands selected by our Method I and WaLuMI for number of bands $K = \{1, 2, 3, 5, 10\}$.

3.3.3. Long-Wavelength Infrared data set. The Long-Wavelength Infrared (LWIR) data set was collected by an interferometer in the $8 - 11$ μm range of the electromagnetic spectrum [40]. During a single scanning, the interferometer collects 20 images from different wavelengths, 256×256 each. Figure 3.11 shows a color image and histogram from one wavelength of a particular data cube. Table 3.6 contains the 20 wavelength numbers at

Table 3.5: Bands selected by Methods I and WaLuMI for the 16-class classification problem.

# Bands, K	**Bands Selected by Method I**	**Bands Selected by WaLuMI [26]**
1	1	129
2	1,34	68,129
3	1,34,2	68,88,129
5	1,34,2,3,29	5,25,68,88,129
10	1,34,2,3,29, 32,41,39,28,42	5,25,100,55,183, 129,79,52,68,88

which the data collection was made. A single data collection event consists of releasing a pre-determined quantity of a chemical liquid into the air to create an aerosol cloud for vapor detection against natural background. The $256 \times 256 \times 20$ cubes are collected successively, i.e., a hyperspectral movie, to record the entire event from 'pre-burst' to 'post-burst'. The three chemicals used in the experiments are Glacial Acetic Acid (GAA), Methyl Salicylate (MeS), and Triethyl Phosphate (TEP). We consider this data as three classes for classification and band selection.

The data was preprocessed using the approach described in [44]. We summarize the approach as follows:

(1) *background estimation:* approximately 50 pre-blast spectral cubes were used and a basis for the background determined for each pixel.

(2) *background removal:* the background was then projected away using the singular value decomposition basis for the background at each pixel;

(3) *k-means clustering:* the resulting background removed pixels were clustered into groups, with each group representing a distinct chemical.

As for the Indian Pines data set analyzed above, we are interested in selecting bands that are the most useful for distinguishing the chemical vapor in airborne plumes. For

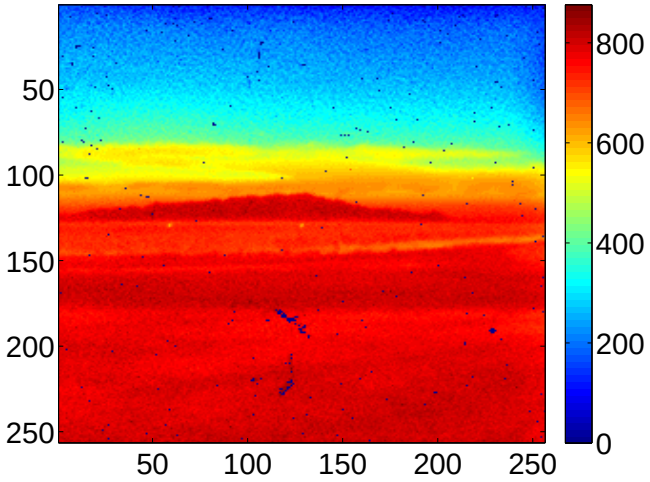
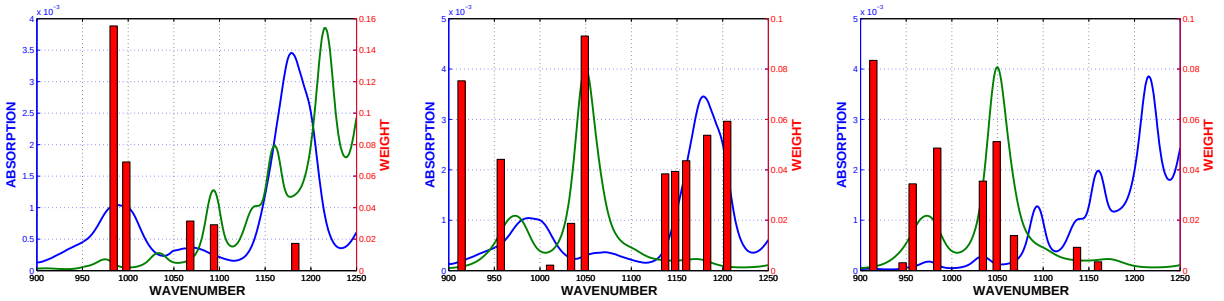


Figure 3.11: An image from one wavelength of a LWIR data cube. Note that the speckling in the image due to the black pixels results from missing measurements where the interferometer was unresponsive. These zero valued pixels were not used in the analysis.

Table 3.6: The LWIR data set wavelengths.

Band index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Wavenumber (cm^{-1})	1011	1034	1049	1068	1094	1107	1137	1148	1160	1183	1205	1216	1237	914	936	946	957	971	984	998

this reason we will focus on applying the SSVM Algorithm to the two class classification problem. We split the 12749 pixels of GAA, 13274 pixels of MeS, and 11986 pixels of TEP in half to obtain training and testing sets. Given the size of the data sets we took 10% of training pixels from each class and sampled randomly with replacement. We used number of bootstraps $N = 100$ and used tolerance equal to 10^{-8} to identify the zero weights at the variability reduction step. The final selection was based on difference in weight magnitudes. The values of C were determined via 5-fold cross-validation on the training data. This data set is clearly linearly separable and the classification results on the test data were essentially perfect. The contribution of this example is the identification of the appropriate bands for the discrimination of these chemicals. The accuracy rates and the band selection results are



(a) (b) (c)

Figure 3.12: Spectral signatures and selected bands for: (a) GAA and MeS, (b) GAA and TEP, (c) MeS and TEP.

shown in Table 3.7. Figure 3.12 depicts plots of spectral signatures combined with selected band weights for the three pairs of the chemicals.

Table 3.7: The LWIR data set: accuracy rates (%) for binary band selection.

Class	**# Bands**	**Bands Selected**	**Accuracy Rate on Reduced Data (%)**
GAA and MeS	5	19,20,4,5,10	100
GAA and TEP	11	3,14,11,10,17,9,8,7,2,1,18	99.9
MeS and TEP	9	14,3,19,2,17,4,7,9,16	99.9

3.4. Summary

We proposed ℓ_1-norm penalized sparse SVMs as an embedded tool for hyperspectral band selection. It is a supervised technique that simultaneously performs band selection and classification. We compared the band selection of the SSVM Algorithm to WaLuMI and Lasso logistic regression for several illustrative classes of the Indian Pines Data Set and compared the bands selected and the classification performance. The SSVM Algorithm selected bands were evaluated using the plot of the difference in spectral curves of the classes. We observed that single bands resided at optimal peaks in these curves. In addition, sets of two or three adjacent bands were selected by the SSVM Algorithm where the slope

of this curve was steep suggesting that multiple bands were needed for sampling. The SSVM Algorithm is trained using bagging to obtain multiple SSVM models and reduce the variability in the band selection. This preliminary band selection is followed by a secondary band selection which involves retraining the SSVM. We used the steep drop in the magnitude of the weights to identify zero weights.

The SSVM Algorithm for binary band selection was extended to the multiclass classification problem using one-against-one (OAO) SSVMs. Three methods were proposed for the multiclass band selection problem. Methods I and II are extensions to the binary band selection; Method III combines a well-known method, WaLuMI, as a preprocessor, with OAO SSVMs. Spatial smoothing by majority filter was used to improve the accuracy rates for different sets of kept bands. Results on both the Indian Pines and the LWIR data sets suggest that the methodology shows promise for both the band selection problem and as a technique that can be combined with other band selection strategies to improve performance.

CHAPTER 4

Classification of Data on Embedded Grassmannians

4.1. Introduction

In this chapter, we pursue classification using comparison between multiple observations of subject classes encoded as linear subspaces. This set-to-set pattern recognition approach captures the signal variability in data. A collection of subspaces has a natural mathematical structure known as the Grassmann manifold (Grassmannian). The Grassmannian is referred to as an *abstract manifold* since it does not reside in Euclidean space, i.e., its properties are not described by n-tuples with the distances between them measured via inner products. Recently it has become an active area of research to develop computational algorithms on non-Euclidean spaces [8, 45, 46].

Nash's famous *isometric embedding theorem* shows under what conditions abstract Riemannian manifolds (Grassmann manifolds are a special case of these), equipped with a Riemannian metric, can be embedded into Euclidean space such that the distances between points on the manifold are preserved [47]. Note that while Nash's isometric (distance-preserving) embeddings exist in general, in this study we consider the existence of an isometric embedding *in the context of metric spaces*, which is not always guaranteed and can be based on the choice of a metric. For instance, according to [48], there is no isometric embedding from any nonempty open subset of the sphere S into any Euclidean space, while the trivial inclusion $S \subset \mathbb{R}^3$ is an isometric embedding of Riemannian manifolds.

A set of points on the Grassmann manifold can be embedded into Euclidean space using projection maps described in [49]. This embedding is isometric if the chordal (projection)

metric is used. We propose an approach for embedding points on the Grassmannian into Euclidean space via multidimensional scaling (MDS), see, e.g., [50] and references therein. The result is a configuration of points in Euclidean space whose Euclidean distances approximate the distances measured on the abstract manifold. The choice of a metric is important, as it changes the geometry of the embedding. For instance, an MDS embedding is isometric if the chordal distance is used on the manifold, while this is not true for other (pseudo)metrics.

Geometric approaches have been proposed for characterizing data on manifolds, i.e., nonlinear objects that behave locally like Euclidean space. These data driven approaches for manifold learning include, e.g., isometric mapping (ISOMAP) [51], local linear embedding (LLE) [52], and Laplacian Eigenmaps [53]. A number of practical algorithms based on ISOMAP and LLE have been proposed for applications to hyperspectral imagery, see, e.g., [54]. We note that in these methods, a manifold coordinate system is derived from computing the geodesic distances between the hyperspectral pixels, i.e., they are algorithms operating in pixel space. The algorithms applied to pixel space are using manifolds as a model for the data. In our approach, that we also illustrate on hyperspectral data, we first encode sets of pixel vectors as subspaces which are viewed as points on a Grassmann manifold, the existence of which is theoretically guaranteed. The Grassmann manifold is then embedded into Euclidean space using MDS. Mapping into Euclidean space is followed by SSVM classification and selection of a subset of dimensions of the embedding based on the sparsity of the SSVM model (Chapter 2). The resulting sparse embeddings, i.e. embeddings with several selected dimensions only, are used for embedded data visualization and model reduction purposes [55].

This chapter has the following outline. In Section 4.2 we describe the mathematical framework behind encoding collections of pixels as subspaces using the geometry of the

Grassmann manifold. In Section 4.3 we outline the methodology for approximating an isometric embedding of the Grassmannian. The algorithm is summarized in Section 4.4 and the experimental results are discussed in Section 4.5. We summarize our findings in Section 4.6.

4.2. The Grassmannian Framework

In the proposed framework, we use the geometric structure of the Grassmann manifold to represent sets of points as subspaces and study the relationship between them.

Definition 4.2.1. *The real* ***Grassmann manifold*** *(Grassmannian) $G(k,n)$ is the manifold of points that parameterize k-dimensional linear subspaces of the real n-dimensional Euclidean space, $\mathbb{R}^n, 0 < k \leq n$.*

The Grassmannian $G(k,n)$ is a compact manifold of dimension $k(n-k)$, and it is a non-Euclidean homogeneous space of the orthogonal group $O(n)$ (that consists of $n \times n$ orthogonal matrices), given by $O(n)/(O(k) \times O(n-k))$ [8]. A point on $G(k,n)$, i.e., a k-dimensional subspace, can be non-uniquely represented by a basis, i.e. an $n \times k$ matrix $\mathbf{U}$ with orthonormal columns ($\mathbf{U}^T\mathbf{U} = \mathbf{I}_k$). Two points on $G(k,n)$ are considered to be the same if they span the same subspace, i.e., notationally, $\mathbf{U}_1 = \mathbf{U}_2$ when $\text{span}(\mathbf{U}_1) = \text{span}(\mathbf{U}_2)$.

To organize original data as points on the Grassmann manifold, we repeatedly sample random k points from the same class to obtain "tall and skinny" matrices $\mathbf{Y}_i \in \mathbb{R}^{n \times k}$, with n being the original data dimension, namely, the number of features. The next step is to compute the reduced singular value decomposition (SVD) $\mathbf{Y}_i = \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^T$, where the $n \times k$ matrix $\mathbf{U}_i$ has orthonormal columns, the $k \times k$ matrix $\mathbf{\Sigma}_i$ is diagonal, and the $k \times k$ matrix $\mathbf{V}_i$ is orthonormal [56]. The $\mathbf{U}_i$ is associated to the column space of $\mathbf{Y}_i$, $\mathcal{R}(\mathbf{Y}_i)$, a k-dimensional subspace of $\mathbb{R}^n$, and thus can represent a point on the Grassmannian, see Figure 4.1.

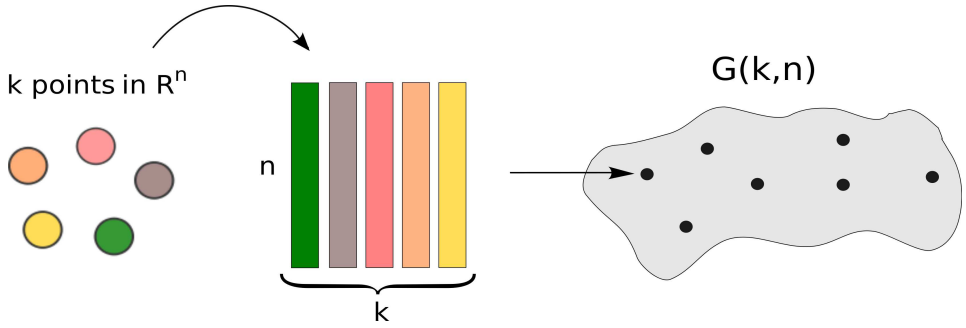


Figure 4.1: Constructing subspaces on a Grassmannian manifold from original data points.

Once the points on $G(k,n)$ are computed, we can generate a matrix of pairwise distances between them. Formally, the Riemannian distance between two subspaces on $G(k,n)$ is the length of the shortest curve connecting them (the geodesic) [45]. There is a way to define distances on the Grassmannian using the principal angles between two subspaces [56].

DEFINITION 4.2.2. *Let* $\mathbf{U}_1$ *and* $\mathbf{U}_2$ *be two orthonormal* $n \times k$ *matrices. The* ***principal angles*** $0 \leq \theta_{min} = \theta_1 \leq \theta_2 \leq \ldots \leq \theta_k = \theta_{max} \leq \pi/2$ *between two subspaces* $span(\mathbf{U}_1) = span(\mathbf{U}_2)$ *are defined recursively by*

$$\theta_i \doteq \underset{\substack{\mathbf{u}_i \in span(\mathbf{U}_1) \\ \mathbf{v}_i \in span(\mathbf{U}_2) \\ \|\mathbf{u}_i\|=\|\mathbf{v}_i\|=1}}{\text{minimize}} \quad \arccos \mathbf{u}_i^T \mathbf{v}_i$$

$$\text{subject to} \quad \mathbf{u}_i^T \mathbf{u}_j = 0, \ \mathbf{v}_i^T \mathbf{v}_j = 0, \ j = 1, \ldots i-1.$$

In practice, the vector of principal angles $\boldsymbol{\theta} \doteq (\theta_1, \theta_2, \ldots \theta_k)$ between two subspaces, given by orthonormal bases $\mathbf{U}_1$ and $\mathbf{U}_2$, can be computed using the SVD [56], see Algorithm 3. Note that the vectors $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ are *principal vectors* between the subspaces spanned by $\mathbf{U}_1$ and $\mathbf{U}_2$.

Let us now define the following distance measures between two subspaces $\mathcal{P}$ and $\mathcal{Q}$ on the Grassmannian (Figure 4.2):

Algorithm 3: Principal Angles

Input: Orthonormal matrices $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{R}^{n \times k}$
$(\mathbf{Y}, \mathbf{\Sigma}, \mathbf{Z}) \leftarrow \text{svd}(\mathbf{U}_1^T \mathbf{U}_2)$
$\boldsymbol{\theta} = \arccos(\text{diag}(\mathbf{\Sigma}))$
Output: Vector of principal angles $\boldsymbol{\theta}$

- The *geodesic* or *arc length* distance is given by

$$d_g(\mathcal{P}, \mathcal{Q}) = (\sum_{i=1}^{k} \theta_i^2)^{1/2} = \|\boldsymbol{\theta}\|_2.$$

- The *chordal* or *projection* distance is given by

$$d_c(\mathcal{P}, \mathcal{Q}) = (\sum_{i=1}^{k} (\sin\theta_i)^2)^{1/2} = \|\sin\boldsymbol{\theta}\|_2.$$

- The third distance is chosen to be a *pseudometric* given by

$$d_l(\mathcal{P}, \mathcal{Q}) = (\sum_{i=1}^{l} \theta_i^2)^{1/2}, l < k,$$

and, in particular, the *smallest principal angle* pseudometric distance is

$$d_1(\mathcal{P}, \mathcal{Q}) = \theta_{min} = \theta_1.$$

Note that d_l is not a metric, as, if $\dim(\mathcal{P} \cap \mathcal{Q}) \geq l$, then $d_l(\mathcal{P}, \mathcal{Q}) = 0$.[1] However, the use of it (and, in particular, d_1) as a distance measure allows for higher accuracy rates in binary experiments for most subspace dimension k values and results in one-dimensional classification models in the case of d_1 [55].

Note that these distance measures lead to different geometries on the Grassmann manifold. In the next section we propose a way to embed $G(k, n)$ into Euclidean space using

[1] E.g., $d_1 = \theta_1$ is zero when two different subspaces intersect at least in a line.

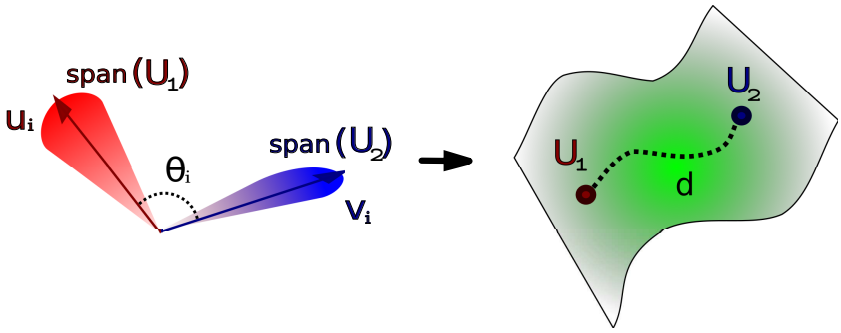


Figure 4.2: Computing principal angles and a distance d between two points on the Grassmannian $G(k,n)$: subspaces $\mathrm{span}(\mathbf{U}_1)$ and $\mathrm{span}(\mathbf{U}_2)$ are represented by orthonormal bases $\mathbf{U}_1$ and $\mathbf{U}_2$.

multidimensional scaling (MDS). We also discuss the resulting MDS embeddings for different geometries and compare them to the well-known projection embedding [49].

4.3. Embedding via MDS

According to [49], the Grassmann manifold $G(k,n)$ can be interpreted as a submanifold of Euclidean space, via the representation of k-dimensional subspaces (given by their bases $\mathbf{U}_i$) by the projection matrices $\mathbf{P}_i = \mathbf{U}_i\mathbf{U}_i^T$. More precisely, if the chordal distance d_c is used, this embedding is isometric (i.e., distance-preserving), and the points are located on a sphere of radius $\sqrt{k(n-k)/n}$ in $\mathbb{R}^N$, with $N = \binom{n+1}{2} - 1 = \frac{n(n+1)}{2} - 1$ and $d_c(\mathbf{U}_1,\mathbf{U}_2) = \frac{1}{\sqrt{2}}\|\mathbf{P}_1 - \mathbf{P}_2\|_F$. Note that N does not depend on k, and becomes very large if the original data has large input space dimension n. To embed points on $G(k,n)$ into Euclidean space of much lower dimension, we propose using multidimensional scaling procedure described, e.g., in [50].

Multidimensional scaling constructs a configuration of points in Euclidean space, only using the information about distances (dissimilarities) between the objects. As the next step of our approach, we use this tool to embed a set of points on the Grassmannian into Euclidean space of the dimension to be determined during the MDS procedure.

Given p points on $G(k,n)$, sampled from raw data, we first generate a symmetric matrix of pairwise distances between the points, $\mathbf{D} \in \mathbb{R}^{p\times p}$, with $\mathbf{D}_{ii} = 0$ and $\mathbf{D}_{ij} \geq 0$, using one of the distance measures introduced in Section 4.2. Then we perform the sequence of steps described in Algorithm 4.

Algorithm 4: Multidimensional Scaling

1 **Input:** Distance matrix $\mathbf{D} \in \mathbb{R}^{p\times p}$
2 Compute $\mathbf{B} = \mathbf{HAH}$, where the centering matrix $\mathbf{H} = \mathbf{I} - \frac{1}{p}\mathbf{e}\mathbf{e}^T$ and $\mathbf{A}_{ij} = -\frac{1}{2}\mathbf{D}_{ij}^2$ ($\mathbf{e}$ is a vector of p ones)
3 Compute $\mathbf{\Gamma\Lambda\Gamma}^T = \mathbf{X}\mathbf{X}^T$, where $\mathbf{X} := \mathbf{\Gamma\Lambda}^{\frac{1}{2}}$
4 **Output:** Configuration of points $\mathbf{X} \in \mathbb{R}^d$, where $d = \text{rank}(\mathbf{B}) = \text{rank}(\mathbf{X})$

Note that since $\mathbf{Be} = 0\mathbf{e}$, then $\mathbf{B}$ always has (at least one) zero eigenvalue corresponding to the eigenvector $\mathbf{e}$. Therefore, $d = \text{rank}(\mathbf{B}) = \text{rank}(\mathbf{X}) \leq p - 1$, i.e., the dimension of the embedding space is never higher than $p - 1$, where p is the number of points on $G(k,n)$.

It can be proved that if $\mathbf{B}$ is positive-semidefinite (i.e., all the eigenvalues of $\mathbf{B}$ are nonnegative), then $\mathbf{D}$ is Euclidean[2] (the converse is also true) [50]. If this is the case, MDS provides an isometric (or distance-preserving) embedding into $\mathbb{R}^d$. If $\mathbf{B}$ is not positive semidefinite, the embedding, using positive eigenvalues of $\mathbf{B}$ only, is adopted as the best approximation we can derive for our non-Euclidean distance matrix $\mathbf{D}$. (Note that in case of small in magnitude negative eigenvalues our loss is little.)

We observe that distances chosen on the Grassmannian provide different MDS embeddings and classification accuracy results (Section 4.5). For instance, the chordal distance between subspaces results in *isometric (distance-preserving)* embeddings for any value of k, while the geodesic and pseudometric distances do not. This observation agrees with the

[2]A distance matrix $\mathbf{D}$ is Euclidean if there exists a configuration of points in some Euclidean space whose interpoint distances are given by $\mathbf{D}$.

results obtained for the projection embeddings in [49]: recall that the representation of k-dimensional subspaces in $\mathbb{R}^n$ by their projection matrices gives a high-dimensional *isometric* embedding of $G(k,n)$ into Euclidean space using d_c. In fact, a configuration obtained by MDS and a configuration obtained via projection matrices using the chordal distance are similar, subject to translation, rotation, and scaling. To show this, one can use *Procrustes analysis* that removes the translational, scaling and rotational components from one configuration so that the optimal alignment between the two embeddings is achieved [50, 57], see Algorithm 5.

Algorithm 5: Procrustes analysis

Input: $\mathbf{X} \in \mathbb{R}^{N_1}$ and $\mathbf{Y} \in \mathbb{R}^{N_2}$, $N_2 \geq N_1$
place $N_2 - N_1$ columns of zeros at the end of matrix $\mathbf{X}$
mean-center both $\mathbf{X}$ and $\mathbf{Y}$ to have the centroids at the origin
find the rotation matrix $\mathbf{A} = \mathbf{H}\mathbf{G}^T$ from $(\mathbf{H}, \mathbf{\Sigma}, \mathbf{G}) \leftarrow \text{svd}(\mathbf{Y}^T\mathbf{X})$
find the scaling factor $\rho = \text{trace}(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})^{1/2}/\text{trace}(\mathbf{X}^T\mathbf{X})$
rotate and scale $\mathbf{X}$ to $\bar{\mathbf{X}} = \rho\mathbf{X}\mathbf{A}$
calculate the Procrustes statistic
$R = 1 - (\text{trace}(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})^{1/2})^2/(\text{trace}(\mathbf{X}^T\mathbf{X})\text{trace}(\mathbf{Y}^T\mathbf{Y}))$
Output: Matched to $\mathbf{Y}$ configuration $\bar{\mathbf{X}}$ and statistic R

We illustrate the similarity of MDS and projection embeddings for the chordal distance d_c via Procrustes analysis on 50 points randomly generated on $G(2,10)$. Figure 4.3 depicts three configurations in Euclidean space, projected on the plane: the MDS embedding $\mathbf{X}$, the projection embedding $\mathbf{Y}$, and the MDS embedding matched to the the projection embedding obtained using Procrustes analysis, $\bar{\mathbf{X}}$. We observe perfect matching between $\bar{\mathbf{X}}$ and $\mathbf{Y}$ with the Procrustes statistic $R = 0$ (meaning the matching is optimal). In contrast to this, MDS and projection embeddings obtained by using distances d_g and d_1 are not matched by Procrustes analysis, see Figures 4.4 and 4.5. This is also justified by nonzero Procrustes

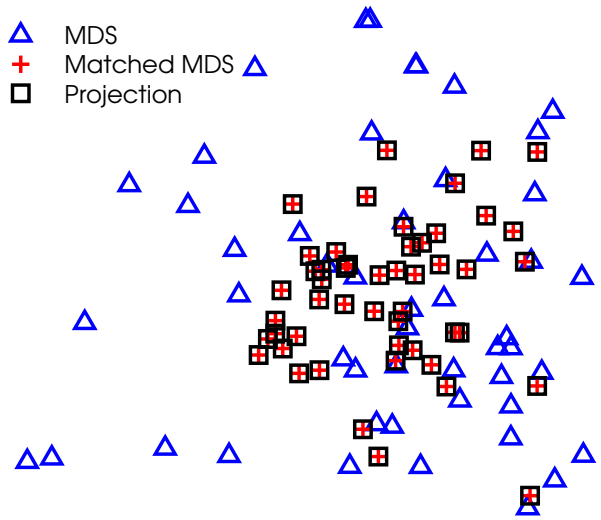


Figure 4.3: Comparison of the MDS and projection embedding configurations obtained from points on $G(2, 10)$ using the chordal distance d_c: MDS configuration $\mathbf{X}$, matched MDS configuration $\bar{\mathbf{X}}$ obtained by Procrustes analysis, and projection configuration $\mathbf{Y}$.

statistic values, which are $R = 0.1215$ for the geodesic and $R = 0.4014$ for the smallest angle distances, respectively.

Based on the analysis above, we conclude that MDS allows for low-dimensional embeddings of $G(k, n)$ into Euclidean space, with distances preserved when using the chordal distance, and distances approximated when using the geodesic or pseudometric distances. An MDS embedding is similar to an embedding via projection matrices, but the latter, in contrast, have significantly higher dimensions given a large enough ambient dimension n. Thus, we adopt MDS for the Grassmannian embedding step, followed by classification and dimension selection in Euclidean space.

4.4. Classification and Dimension Selection

Once we have obtained a configuration of points in d-dimensional Euclidean space, we assign each point a class label from the original class of pixels, as the way the point is

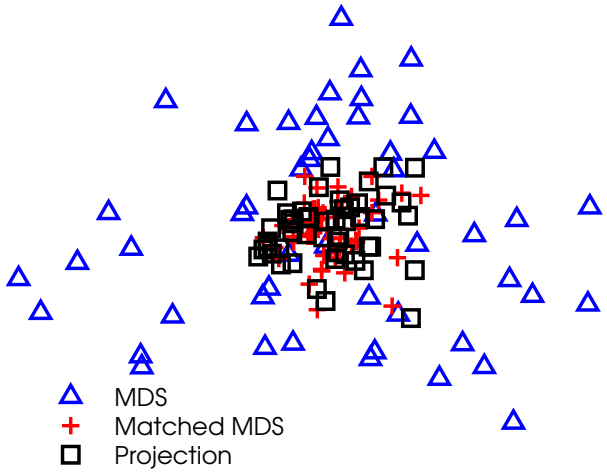


Figure 4.4: Comparison of the MDS and projection embedding configurations obtained from points on $G(2, 10)$ using the geodesic distance d_g: MDS configuration $\mathbf{X}$, matched MDS configuration $\bar{\mathbf{X}}$ obtained by Procrustes analysis, and projection configuration $\mathbf{Y}$.

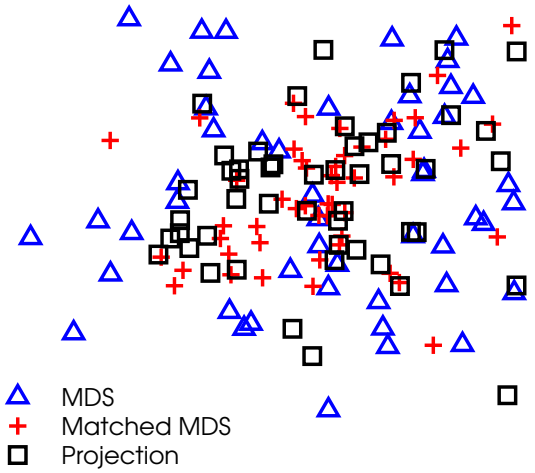


Figure 4.5: Comparison of the MDS and projection embedding configurations obtained from points on $G(2, 10)$ using the smallest principle angle distance d_1: MDS configuration $\mathbf{X}$, matched MDS configuration $\bar{\mathbf{X}}$ obtained by Procrustes analysis, and projection configuration $\mathbf{Y}$.

Algorithm 6: Dimension Selection

Input: Configuration of labeled points $\mathbf{X} \in \mathbb{R}^d$ (embedding space)
Train an SSVM model $\rightarrow$ weight vector $(w_1, w_2, \ldots, w_d)^T$
Rank the weights by magnitude: $(w_{i_1}, w_{i_2}, \ldots, w_{i_d})^T$ such that $|w_{i_1}| \geq |w_{i_2}| \geq \ldots \geq |w_{i_d}|$
If $|w_{i_k}|/|w_{i_{k+1}}| = O(10^M)$ and $M > 1$ for some $k = k^*$, remove dimensions starting from index i_{k*+1}
Output: Selected dimensions

computed allows one to retain this information. We can now perform classification and dimension selection for further study and model reduction by training a sparse support vector machine (SSVM), refer to Chapter 2.

Recall that the SSVM is a supervised classification method that seeks for the optimal separating hyperplane between two classes of data, and builds a sparse model due to the ℓ_1-norm regularization term in the objective function. The sparsity of the weight vector in the decision function, can be used to reduce the number of data features. Note that in our case, a set of features to be reduced by the SSVM is *a set of d dimensions* of the embedding space $\mathbb{R}^d$. In general, feature selection reduces the size of the data, and, consequently, the computational cost for further experiments, improves classification rates, or eliminates redundant features. In our case, the optimal dimensions determined by the SSVM can be used for model reduction and embedding visualization, which we demonstrate in Section 4.5. The approach for selecting embedded dimensions based on the SSVM is summarized in Algorithm 6.

Table 4.1 illustrates how this works in practice. The dimension selection results are given for two classes of the AVIRIS Indian Pines data set [38], Corn-notill and Grass/Pasture, encoded on $G(10, 220)$ and embedded into $\mathbb{R}^{199}$ using the distances d_1 and d_c. In both cases there is a gap in between "important" dimensions corresponding to heavier weights and

Table 4.1: Two classes of the AVIRIS Indian Pines data set, Corn-Notill and Grass/Pasture: SSVM dimension selection of MDS embedding space using d_1 and d_c distances on $G(10, 220)$:

pseudometric d_1		**chordal** d_c	
Dimension	Weight	Dimension	Weight
1	4.1658e+01	1	4.4606e+00
2	3.9670e-08	10	6.7933e-01
3	8.6623e-09	29	2.6502e-01
12	8.2808e-09	82	2.0162e-01
20	7.7610e-09	30	6.3833e-02
22	7.1066e-09	8	2.7234e-02
14	7.0018e-09	74	1.2370e-06
...	...	...	...

the dimensions to be eliminated, determined by Algorithm 6. In particular, dimension 1 is selected if d_1 is used, and dimensions (1,10,29,82,30,8) are selected if d_c is used.

Our approach is summarized in Algorithm 7.

Algorithm 7: Classification on Embedded Grassmannians

1. From original data points in $\mathbb{R}^n$, compute p points on $G(k, n)$ for chosen k and p
2. Compute pairwise distances between the points (e.g., d_c, d_g, or d_1)
3. Embed points on $G(k, n)$ into $\mathbb{R}^d$ via MDS (Algorithm 4)
4. Train an SSVM and select dimensions in $\mathbb{R}^d$ (Algorithm 6)

The SSVM is a binary classifier, so in case of $c > 2$ data classes, we realize an embedding by MDS using a distance matrix D that contains pairwise distances between all the points from different classes. Using one-against-one (OAO) SSVM approach (see Section 3.2.2), we can classify c classes in $\mathbb{R}^d$ by training $\binom{c}{2} = \frac{c(c-1)}{2}$ binary models, and then applying majority voting to assign class labels to testing points. Note that the dimension of the embedding in the multiclass case is much higher than in the binary case, provided we compute the same number of points for each class. The pairwise multidimensional scaling is not applicable, as the resulting $\frac{c(c-1)}{2}$ two-class embedding spaces have different dimensions.

4.5. Experimental Results

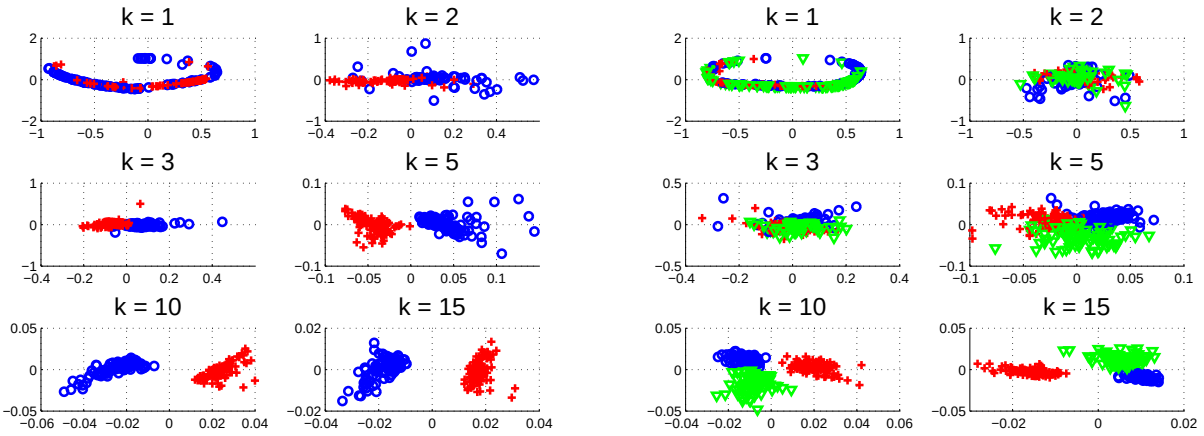
We apply our method to classification of labeled hyperspectral imagery. The experimental results are obtained on the Indian Pines and Pavia University data sets.

4.5.1. AVIRIS Indian Pines Data Set. This data set has been described in detail in Section 3.3.2. Note that for this data set, the Grassmannian is $G(k, 220)$, where k is the dimension of subspaces to be chosen, and $n = 220$ is the ambient (pixel) dimension. For a typical experiment, we constructed 100 subspaces per class, with 50 for training and 50 for testing. We have found this optimal number experimentally, by training SSVMs on different number of points embedded into Euclidean space, using the chordal, the geodesic, and the smallest principal angle distances.

By realizing MDS embeddings of $G(k, n)$ under different distances frameworks, we observed that the chordal distance d_c provided distance-preserving embeddings, while the geodesic distance d_g and the pseudometric d_1 resulted in no isometry.[3] Recall that k-dimensional subspaces in $\mathbb{R}^n$ can also be embedded isometrically into $\mathbb{R}^N$, with $N = \binom{n+1}{2} - 1$, via a projection embedding using the chordal distance d_c. For $n = 220$, N becomes $\binom{221}{2} - 1 = 24309$, while MDS embeds $G(k, 220)$ into $\mathbb{R}^d$, where $d \leq p - 1 = 199$, provided we have $p = 200$ points on $G(k, 220)$ sampled from the original data. It is worth mentioning that the isometry under the chordal distance framework did not necessarily result in the best classification models. In fact, we found that for some k values, the d_1 distance framework provided the highest accuracy rates in two-class experiments.

Figure 4.6 illustrates configurations of points in Euclidean space obtained by embedding points on $G(k, 220)$ via MDS under the d_1 framework, for various subspace dimension values

[3]Recall that the diagnostic for isometry is the spectrum of the MDS matrix $\mathbf{B}$. If there are no negative eigenvalues present (i.e., $\mathbf{B}$ is positive-semidefinite) then the distance matrix $\mathbf{D}$ is Euclidean.



(a) (b)

Figure 4.6: Pseudometric d_1 embeddings of $G(k, 220)$ via MDS for the Indian Pines data set classes for various k (the two dimensions correspond to the top eigenvectors of $\mathbf{B}$): (a) Corn-notill (o) versus Grass/Pasture (+); (b) Corn-notill (o), Soybeans-notill (+), and Soybeans-min ($\triangle$).

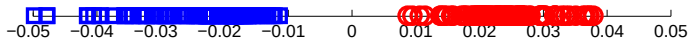
k. Here we have examples for two and three classes of the Indian Pines data set. Note that the two-dimensional representation of the configurations is obtained by using two dimensions corresponding to the top eigenvalues of the matrix $\mathbf{B}$. We see the classes separation becoming stronger as we increase the dimension k of the subspaces.

Table 4.2 shows results for two-class experiments on $G(10, 220)$: classes Corn-notill vs. Grass/Pasture and Soybeans-notill vs. Soybeans-min. As we have mentioned before, only the chordal distance provides isometric embeddings (the number of negative eigenvalues of $\mathbf{B}$ is zero). However, the best SSVM accuracy rates are obtained by using the pseudometric d_1. Note that the classes Soybeans-notill and Soybeans-min are separated with 100% accuracy, which is known to be the best result for this high difficulty classification problem, see also [55].

The sparse SVM selects the optimal dimensions of embedding spaces. For the chordal and geodesic distances, we obtain different combinations of selected embeddings. The use of the

Table 4.2: Two-class experiments for the Indian Pines data set: $p = 200$ points on $G(10, 220)$. The results are averaged over 10 runs.

Classes	**Number of negative eigenvalues of B**			**SSVM Accuracy (%)**			**Number of dimensions selected**		
	d_c	d_g	d_1	d_c	d_g	d_1	d_c	d_g	d_1
Corn-notill vs. Grass/Pasture	0	3.9	95.9	100	100	100	1.1	3.9	1
Soybeans-notill vs. Soybeans-min	0	2.7	91.3	87.2	74	100	24.5	46.9	1



(a)

(b)

Figure 4.7: Two-class pseudometric d_1 embeddings of $G(10, 220)$ using one dimension selected by the SSVM for: (a) Corn-notill (□) and Grass/Pasture (o) classes; (b) Soybean-min (o) and Soybeans-notill (□) classes.

pseudometric d_1 in our framework resulted in one selected dimension for both experiments in Table 4.2, which can be used as a projection direction to visualize the embedded data separation, see Figure 4.7. An interesting observation from our experiments is that for the pseudometric framework our algorithm always selected the first dimension of an embedding corresponding to the first principal direction of MDS with the largest eigenvalue of the MDS matrix $\mathbf{B}$.

More results on SSVM accuracy, as a function of subspace dimension k, are given in Figure 4.8. First, we note that different geometry of the three $G(k, 220)$ frameworks results in different functions as k grows. Second, the d_1 framework overperforms the other two (chordal

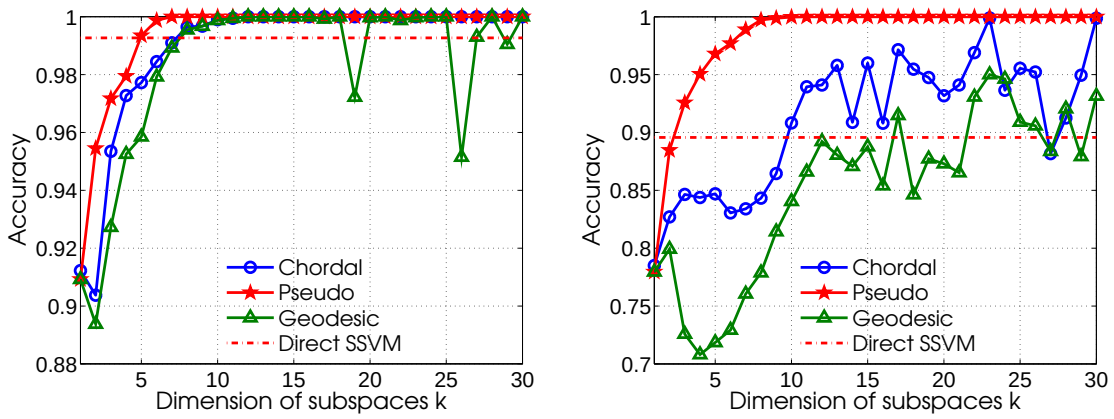


Figure 4.8: SSVM accuracy as a function of k for the Indian Pines data set for chordal, geodesic, and pseudometric d_1 frameworks on $G(k, 220)$. Comparison with (direct) SSVM accuracy obtained on the original data points for: (a) Corn-notill and Grass/Pasture; (b) Soybeans-notill and Soybean-min. (Results are averaged over 10 runs.)

and geodesic) for both low difficulty (Corn-notill and Grass/Pasture) and high difficulty (Soybeans-notill and Soybean-min) classification tasks, as well as the direct applications of SSVMs to the original data points.

As described in Section 4.4, in case of more than two classes, we realize configuration of points in Euclidean space by embedding all the subspaces from different classes at one setting, using a matrix that contains pairwise distances between all the points on $G(k, 220)$. Figure 4.9 shows accuracy rate versus subspace dimension k for nine-class experiments[4] using different distances: chordal d_c , geodesic d_g, and pseudometrics $d_1 = \theta_1$, $d_2 = (\theta_1^2 + \theta_2^2)^{1/2}$ and $d_3 = (\theta_1^2 + \theta_2^2 + \theta_3^2)^{1/2}$. The plots reflect the difference in the geometry of the frameworks. For instance, as we increase k in $G(k, 220)$, the pseudometric d_1 will be zero or close to zero for most of the data, due to the high concentration of subspaces on the manifold, causing decrease in classification rates. The other measures are more discriminative as k grows,

[4]Classes included: Corn-notill, Corn-min, Grass/Pasture, Grass/Trees, Hay-windrowed, Soybeans-notill, Soybeans-min, Soybeans-clean, and Woods.

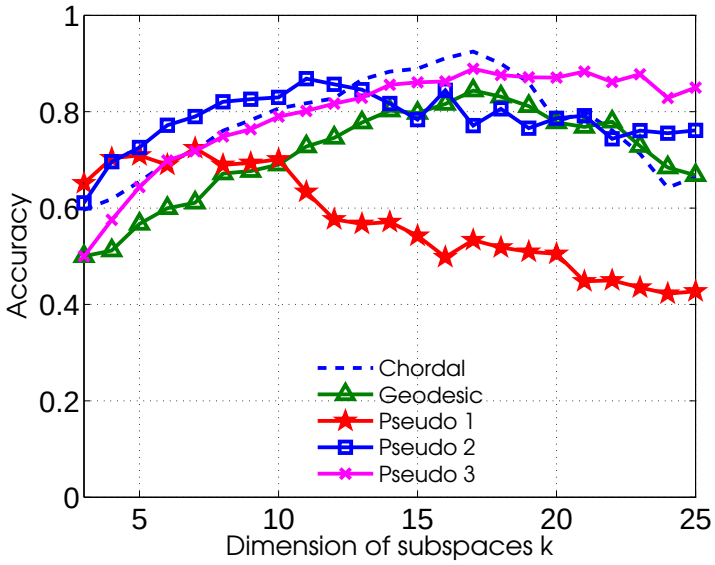
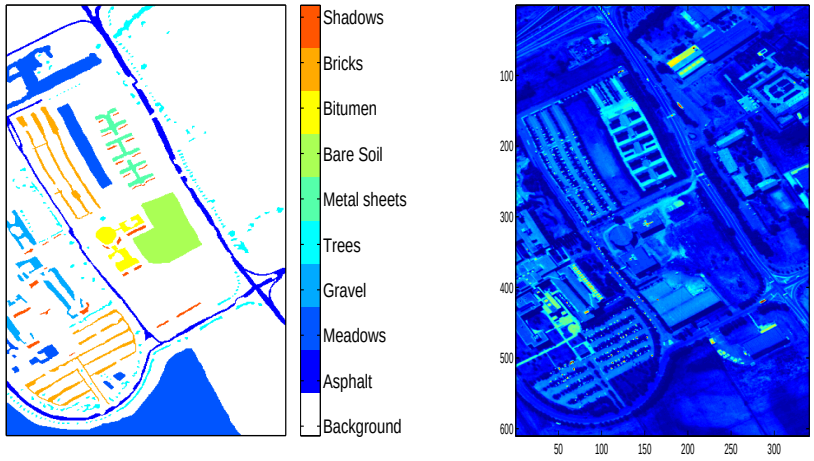


Figure 4.9: SSVM accuracy as a function of k for nine classes of the Indian Pines data set, using chordal d_c , geodesic d_g, and pseudometric distances d_1, d_2 and d_3 on $G(k, 220)$. (Results are averaged over 10 runs.)

compare, e.g., d_1 and d_2: the use of even two principal angles in the pseudometric results in better performance starting from $k = 5$.

4.5.2. Pavia University Data Set. This hyperspectral data set was collected by the Reflective Optics Spectrographic Imaging System (ROSIS) imaging spectrometer over the urban area of the University of Pavia, Italy [58]. The image size in pixels is 610×340, and the number of spectral bands is 103, with spectral range from 0.43 to $0.86\mu m$. Note that for this data set, the Grassmannian becomes $G(k, 103)$, where k is a subspace dimension parameter. Figure 4.10 shows the nine reference classes of interest and one band image. As the previous data set, this data was also mean-centered and randomly partitioned into 50% for training and 50% for testing.

Table 4.3 contains typical binary results for two pairs of classes, Asphalt vs. Trees and Asphalt vs. Gravel, on embedded $G(k, n)$. Similar to the Indian Pines data set binary experiments, we observe that the pseudometric d_1 framework results in better classification



(a) (b)

Figure 4.10: ROSIS Pavia University data set: (a) ground truth; (b) one band image.

Table 4.3: Two-class experiments for the Pavia University data set: $p = 200$ points on $G(10, 103)$. The results are averaged over 10 runs.

Classes	**Number of negative eigenvalues of B**			**SSVM Accuracy (%)**			**Number of dimensions selected**		
	d_c	d_g	d_1	d_c	d_g	d_1	d_c	d_g	d_1
Asphalt vs. Trees	0	29.1	94.6	100	100	100	1	1.7	1
Asphalt vs. Gravel	0	25	93.6	91.3	83.3	100	22.5	57.3	1

rates, and one-dimension SSVM-based selection in embedding spaces. Isometric embeddings were obtained using the chordal distance framework.

Choosing higher subspace dimensions k does not necessarily results in better prediction, depending on the geometry of the framework. For instance, Figure 4.11 shows that when $k > 15$, the smallest angle distance $d_1 = \theta_1$ stops being discriminative. Note that for $k \leq 15$, in both pairs, Asphalt and Gravel (high difficulty classification case) and Asphalt and Trees (low difficulty classification case), d_1 framework overperforms the other two (d_g and d_c). Thus, d_1 can be robust, but on the other hand, if we increase k too high, the geometry of

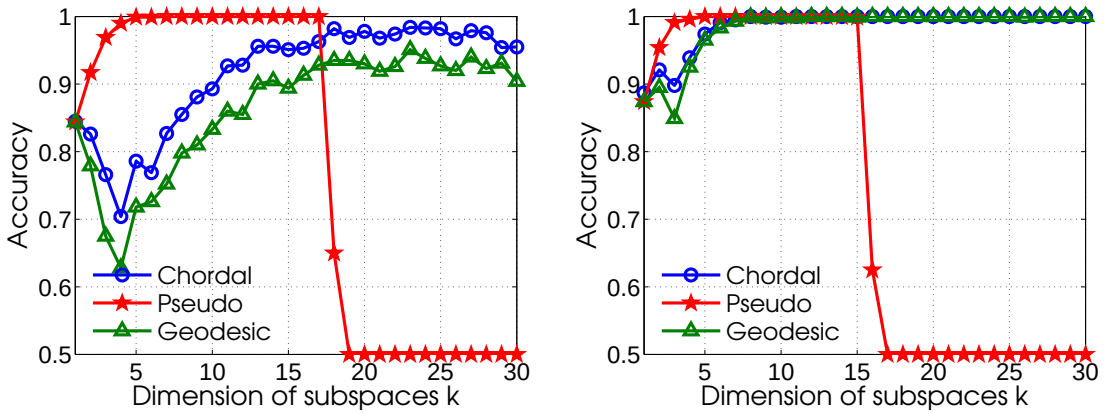


Figure 4.11: SSVM accuracy as a function of k for the Pavia University data set classes for chordal, geodesic, and pseudometric d_1 frameworks on $G(k, 103)$: (a) Asphalt and Gravel; (b) Asphalt and Trees. (Results are averaged over 10 runs.)

the manifold may change such that the smallest angle distances become close or equal to zero for many subspaces.

Figure 4.12 contains plots of accuracy as a function of k for all the nine classes of the Pavia University data set. By varying k in the $G(k, 103)$ settings, we compare SSVM results under d_c, d_g, d_1, d_2, and d_3 frameworks that have different geometry. We notice that the smallest angle distance d_1 framework outperforms the other ones for smaller k values, but it becomes non-discriminative starting from $k = 15$. We observe that including more principal angles in a distance measure results in better SSVM performance, as k grows (e.g., compare plots for the pseudometrics d_1, d_2 and d_3). The interpretation is the following: by including more original points in a subspace, we make the points on the Grassmannian share more information, and as a result, we need more principal angles between the subspaces to discriminate between them.

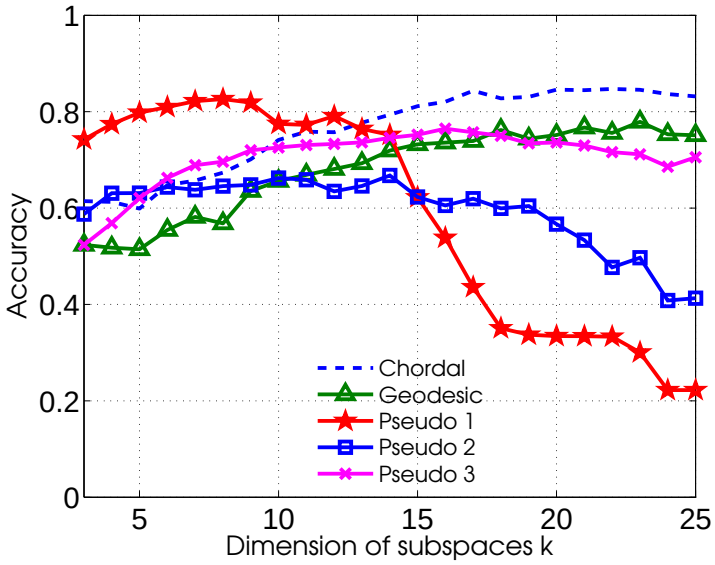


Figure 4.12: SSVM accuracy as a function of k for nine classes of the Pavia University data set, using chordal d_c , geodesic d_g, and pseudometric distances d_1, d_2 and d_3 on $G(k, 103)$. (Results are averaged over 10 runs.)

4.6. Summary

The proposed approach shows how to take raw data (generally not on a manifold) and encode it on the Grassmannian $G(k, n)$, enabling the exploitation of a rich geometric framework. We observed that the smallest principal angle pseudometric provided the best classification accuracy in our binary experiments, for particular k values, including the high difficulty classification pairs of classes of both data sets. We note that in some experiments under the d_1 framework, higher k values did not result in better prediction, meaning that the subspaces intersect at least in a line, forcing the smallest angle distance to be zero. SSVMs effect sparse dimension selection for optimal binary classification, even as low as one-dimension for pseudometric d_1 embeddings. We observed that only the chordal distance provides isometric embeddings which agrees with [49].

In case of $c > 2$ classes of data, we realize an "all-in-one" embedding, by forming a distance matrix $\mathbf{D}$ from pairwise distances between all the points constructed on $G(k, n)$

from different classes. Note that although this increases the dimension of the embedding, pairwise MDS results in $\frac{c(c-1)}{2}$ embeddings that differ in dimension sizes, therefore making an application of OAO SSVM impossible. An interesting observation we have made is that for bigger k's, the smallest principle angle pseudometric may be less discriminative compared to the other distances that include two or more principal angles. High-dimensional subspaces may have zero smallest principal angle for most of the data, due to intersection occurring between them.

Future work may include comparison of points on $G(k, n)$ to points $G(j, n)$ where $k \neq j$. Also, it would be interesting to use a patch-based approach for constructing subspaces on the Grassmannian: instead of sampling the set of pixels randomly from each class, it can be done by taking the points that are close to each other, i.e. lying in some neighborhood "patch".

CHAPTER 5

An Application of Persistent Homology on Grassmann Manifolds for the Detection of Signals in Hyperspectral Imagery

5.1. Introduction

In this chapter, we present an application of persistent homology to the detection of chemical plumes in hyperspectral imagery [59]. Recall that a digital hyperspectral image is a three-dimensional array consisting of two spatial dimensions and one spectral dimension, called a data cube, see Figures 5.1 and 3.1. Including a temporal dimension in the process of data acquisition provides dynamic hyperspectral information in a four-way array. Such sequence of hyperspectral cubes collected at short time intervals is effectively a hyperspectral movie capturing potentially interesting spectral changes in a scene such as the release of a chemical plume. An important application of dynamic hyperspectral imaging is in the surveillance of the atmosphere for chemical or biological agents [60].

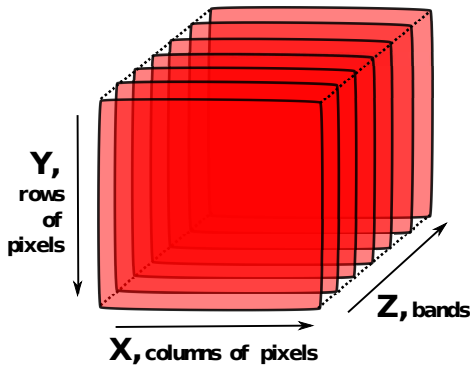


Figure 5.1: Hyperspectral data cube.

Persistent homology (PH) is a relatively new tool in topological data analysis (TDA) that provides a multiscale method for analyzing the topological structure of data sets [9, 10]. The direct application of PH to large data sets, such as sequences of hyperspectral data cubes, can be prohibitive due to computational intractability. We overcome this issue by encoding the frames of a hyperspectral movie as points on a Grassmann manifold [8]. Recall that the real Grassmannian provides a parameterization of k-dimensional linear subspaces of $\mathbb{R}^n$ and a geometric framework for the representation of a set of raw hyperspectral data points by a single manifold point (Section 4.2). This approach affords a form of compression while retaining pertinent topological structure. In this setting, it becomes feasible to utilize PH to analyze larger volumes of hyperspectral data as the high computational cost of PH applied to the original data space is greatly reduced.

We apply this approach to the detection of chemical signals in the collection of data cubes of the Long-Wavelength Infrared (LWIR) data set [40]. Under the proposed framework, raw data cubes are mapped into a Grassmann manifold, and, for a particular choice of a distance metric, it is possible to generate topological signals that capture changes in the scene after a chemical release.

This chapter is organized in the following order: Section 5.2 describes PH, while the Grassmannian framework is explained in Section 5.3. Computational experiments are discussed in Section 5.4, followed by summary in Section 5.5.

5.2. Persistent Homology

Persistent homology (PH) is a computational approach to topology that allows one to answer basic questions about the structure of point clouds in data sets [9, 10]. This procedure involves interpreting a point cloud as a noisy sampling of a topological space. Aspects of

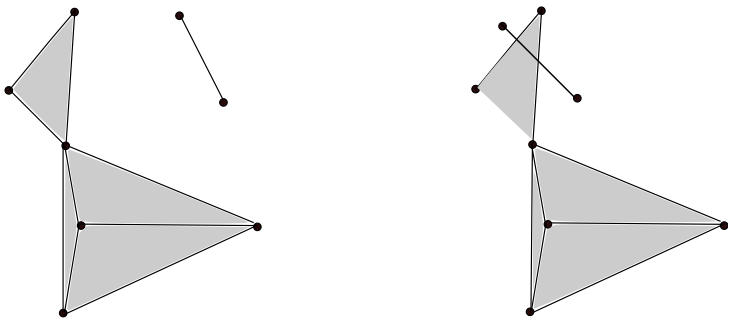


Figure 5.2: Examples of a simplicial complex (left) and a non-simplicial complex (right).

this topological space are uncovered by associating, to the data cloud, a nested sequence of simplicial complexes indexed by a scale parameter ϵ. A simplicial complex is a finite set of k-simplices (simple pieces). A *k-simplex* is defined as the convex hull of $k+1$ points in $\mathbb{R}^n$. For instance, a 0-simplex is a vertex, a 1-simplex is an edge, a 2-simplex is a triangle, and so on. A *face* of a k-simplex is a lower dimensional simplex belonging to the k-simplex.

DEFINITION 5.2.1. *A **simplicial complex** S in $\mathbb{R}^n$ is a collection of simplices such that:*

- *every face of a simplex in also belongs to S;*
- *the intersection of any two simplices in S is a face of each of them.*

It follows that for a simplicial complex, two k-simplices either intersect in a face or are disjoint. See examples of a simplicial and non-simplicial complexes in Figure 5.2.

The *Vietoris-Rips* complex (or the *Rips* complex) is one of the methods used in PH procedure [61]. To build such a complex, one starts from a matrix of pairwise distances between points in the cloud. Given a scale parameter $\epsilon > 0$, a simplicial complex $S(\epsilon)$ is constructed in such a way that every set of $k+1$ points forms a k-simplex if the the pairwise distances between the points is less than ϵ. Figure 5.3 illustrates the construction of ϵ-dependent Rips complexes from a finite set of points. The connectivity of a simplicial

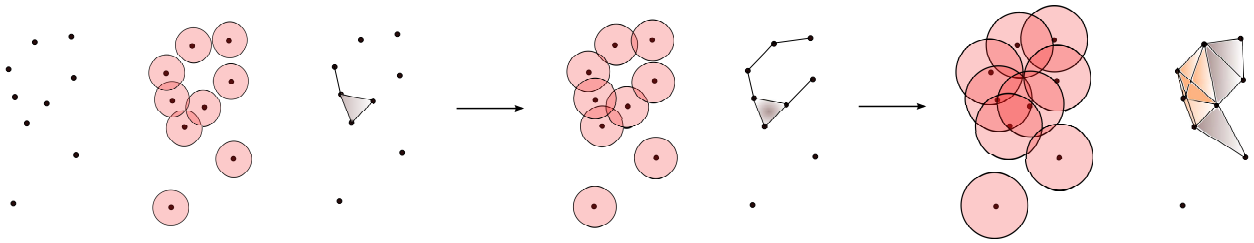


Figure 5.3: Three Rips complexes build from a finite set of points using different ϵ values.

complex may be viewed as arising from the overlapping of ϵ-balls that cover the data in the point cloud.

Of particular interest are ϵ-dependent, kth order holes in a simplicial complex, for these provide insight into the topological structure at different scales. For instance, zeroth order holes give the number of connected components (clusters) of the point cloud, while first order holes indicate the existence of topological circles, or periodic phenomenon. A tool from algebraic topology, *homology*, uncovers kth order holes in a simplicial complex, by encoding the topological information into an algebraic form [62]. In particular, to compute homology for a given simplicial complex $S(\epsilon)$ and $k > 0$, an abstract vector space C_k is generated, with basis consisting of the set of k-simplices in $S(\epsilon)$. The dimension of C_k is equal to the number of k-simplices. The elements of C_k are called *k-chains*.

The boundary of a k-simplex is the union of the $(k-1)$-faces belonging to the simplex. By defining boundary operators $\partial_k : C_k \to C_{k-1}$, one can connect the vector spaces C_k into a *chain complex*:

$$\cdots \to C_{k+1} \overset{\partial_{k+1}}{\to} C_k \overset{\partial_k}{\to} C_{k-1} \to \cdots \to C_2 \overset{\partial_2}{\to} C_1 \overset{\partial_1}{\to} C_0 \overset{\partial_0}{\to} 0.$$

Each C_k has two important subspaces, namely:

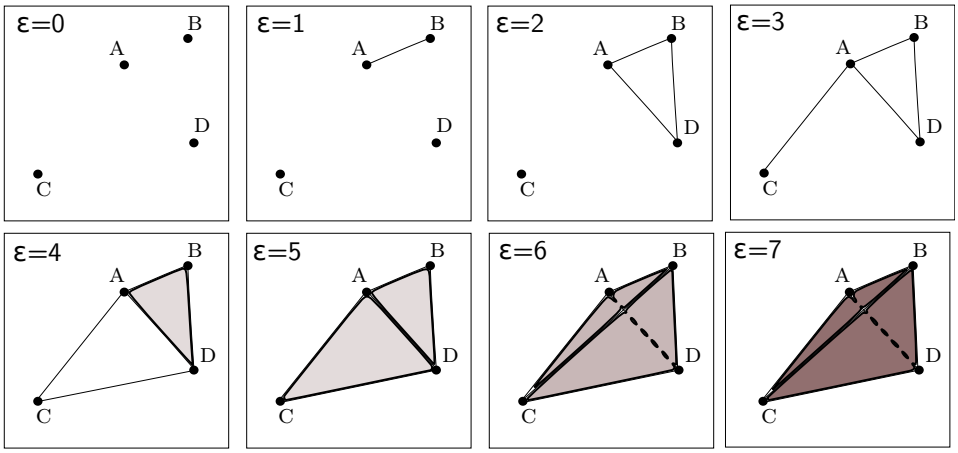
- k-cycles: $Z_k \doteq \ker(\partial_k : C_k \to C_{k-1})$,
- k-boundaries: $B_k \doteq \text{im}(\partial_{k+1} : C_{k+1} \to C_k)$.

Note that $\partial_k \circ \partial_{k+1} = 0$, i.e., a boundary has no boundary. It can be shown that this equation is equivalent to the following inclusion: $B_k \subseteq Z_k \subseteq C_k$ [62].

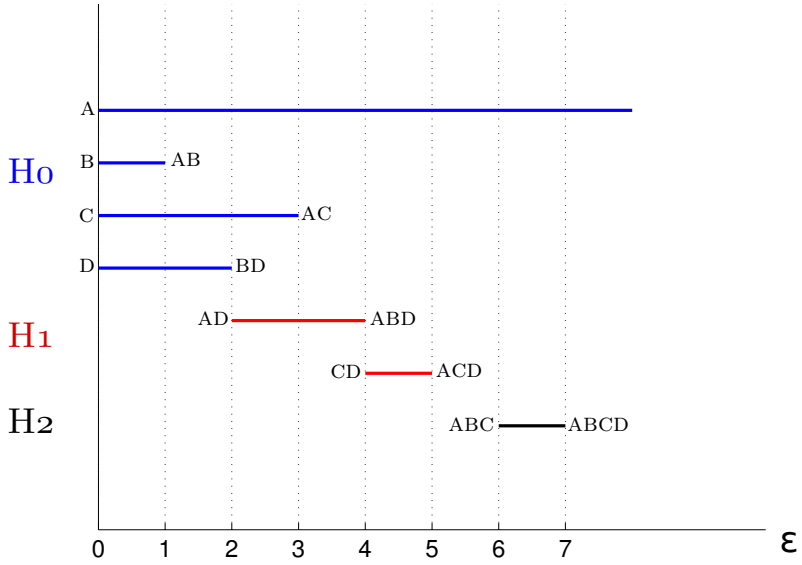
The kth simplicial homology group of the chain complex is defined to be the quotient group $H_k = Z_k/B_k$. This group is made up of classes of k-cycles, where two k-cycles are in the same class (i.e., *homologous*) if their difference is a boundary. The *kth Betti number*, $\beta_k = \dim(H_k) = \dim(Z_k) - \dim(B_k)$, the rank of the associated kth homology group of the simplicial complex, equals the number of k-dimensional holes [62].

To convert a point cloud data set into a simplicial complex, a choice of ϵ is required. In persistent homology, one seeks structures that persist over a range of scales, rather than looking for an optimal choice for ϵ [61]. PH tracks homology classes of the point cloud along the scale parameter, building an inclusion of simplicial complexes $S(\epsilon_1) \subseteq S(\epsilon_2) \subseteq \ldots \subseteq S(\epsilon_N)$ and indicating at which ϵ a hole appears and for which range of ϵ values it persists. The Betti numbers, as functions of the scale ϵ, are visualized in a distinct barcode for each dimension k [61].

Figure 5.4 schematically illustrates the Rips complexes of 4 points generated for different ϵ values and the corresponding $Betti_0$, $Betti_1$, and $Betti_2$ barcodes. In the barcode (Figure 5.4b), the horizontal axis corresponds to ϵ values, while the vertical axis depicts arbitrarily ordered homology classes of dimension k. Each horizontal bar represents the birth-death of a topological feature. The kth Betti number at any ϵ value is the number of bars that intersect the vertical line through ϵ. For instance, at $\epsilon = 0$ we have 4 isolated points or clusters (A,B,C,D), i.e., $Betti_0 = 4$, and at $\epsilon = 2$ we have two clusters (point C and triangle ABD), i.e., $Betti_0 = 2$, and a topological circle (ABD) with $Betti_1 = 1$.



(a)



(b)

Figure 5.4: Example of PH barcode generation: (a) the Rips complexes of 4 points for different scale ϵ values; (b) the corresponding $Betti_0$, $Betti_1$, and $Betti_2$ barcodes displayed with the blue, red, and black bars, respectively.

Figure 5.5 shows an example of the $k = 0$ and $k = 1$ barcodes generated for a point cloud sampled from the unit circle. We conclude that $Betti_0 = Betti_1 = 1$ which corresponds to the number of connected components and number of loops, respectively, shown by the longest (persistent) horizontal bars in each plot.

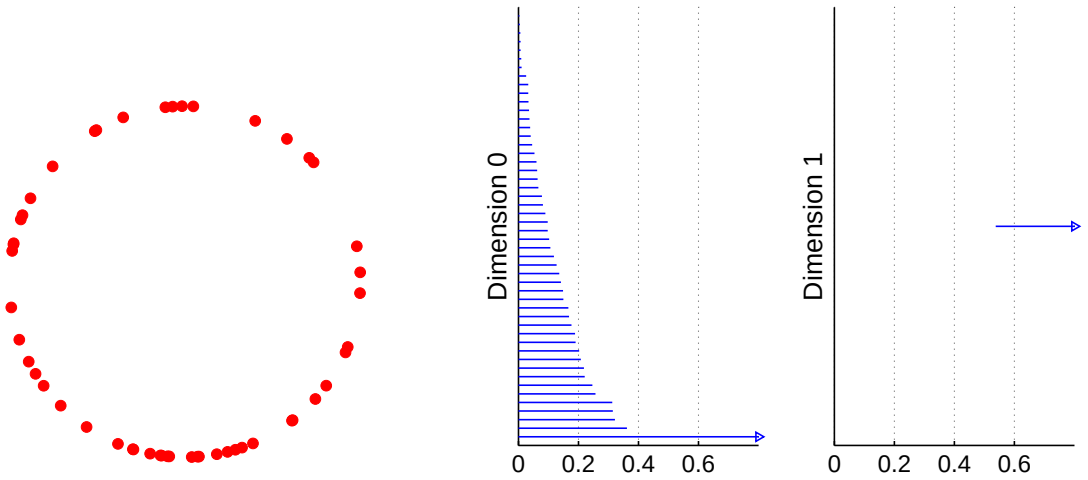


Figure 5.5: $Betti_0$ and $Betti_1$ barcodes (right) corresponding to point cloud data sampled from the unit circle (left).

A two thousand point cloud sampled from a three-dimensional torus has $k = 0$, $k = 1$, and $k = 2$ barcodes shown in Figure 5.6. From these, we conclude that $Betti_0 = Betti_2 = 1$ and $Betti_1 = 2$ (each corresponding to the number of persistent bars in the barcode) which agrees with the fact that a torus has one connected component, two circular holes, and a two-dimensional void.

To generate the barcodes, we use JavaPlex, a library for persistent homology and topological data analysis [63]. In the next section, we discuss how PH can be used for HSI signal detection.

5.3. The Grassmannian Framework

Similar to the previous chapter, we propose using the Grassmann manifold (Grassmannian) as a framework, but now for detection of signals in hyperspectral imagery via PH. (Section 4.2 contains the background material on the Grassmann manifold and its geometry.) This framework enables the processing of large data sets, such as the hyperspectral

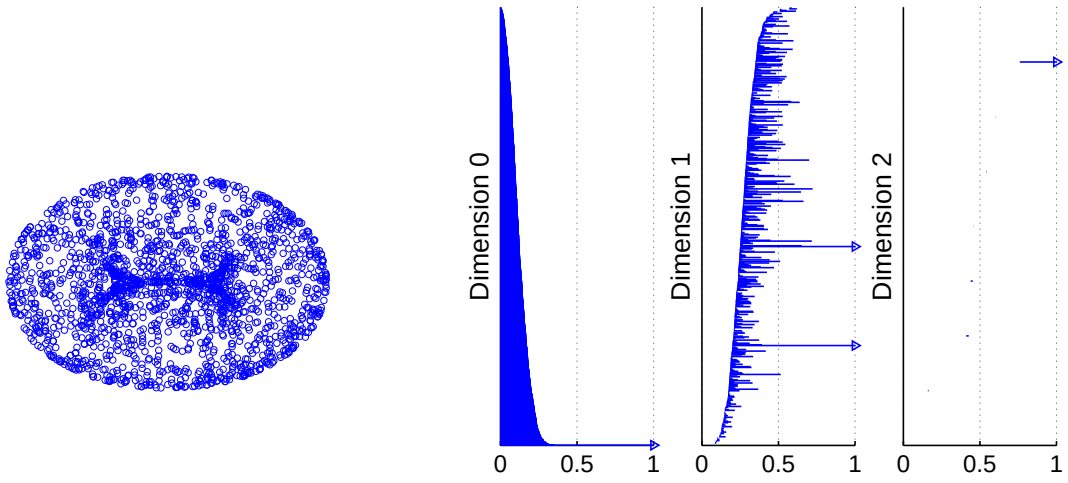


Figure 5.6: $Betti_0$, $Betti_1$, and $Betti_2$ barcodes (right) corresponding to point cloud data sampled from a three-dimensional torus (left).

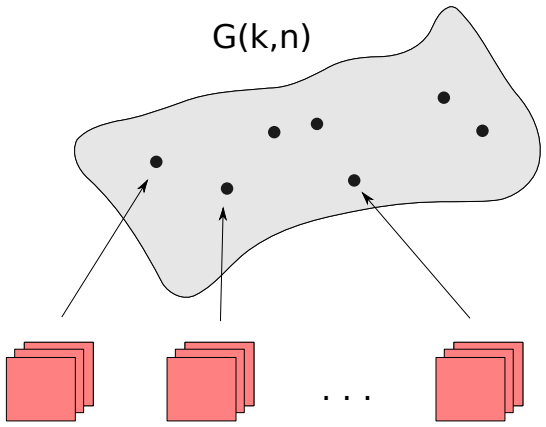


Figure 5.7: A sequence of data cubes mapped to points on $G(k, n)$.

movies explored in this investigation, while retaining valuable discriminative information. Recall that the real Grassmann manifold $G(k, n)$ is the collection of all k-dimensional subspaces of the vector space $\mathbb{R}^n$ [8]. A sequence of hyperspectral data cubes, or subcubes taken from them, can be mapped to points on $G(k, n)$. Figure 5.7 schematically illustrates the setting.

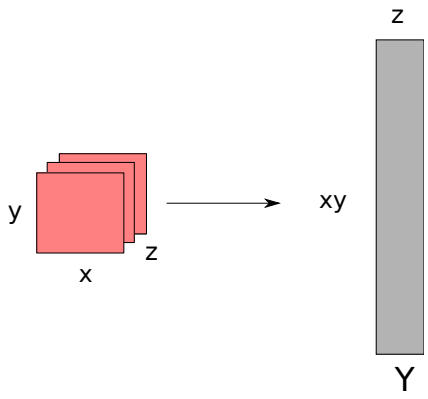


Figure 5.8: An xyz-cube reshaped into an $xy \times z$ matrix $\mathbf{Y}$ ($z < xy$).

Given a xyz-cube, one can reshape it into an $xy \times z$ matrix $\mathbf{Y}$, whose columns span a subspace on $G(k, n)$ with $k = z$ and $n = xy$, provided $z < xy$, see Figure 5.8.

If we compute the reduced SVD, $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, the columns of the $n \times k$ orthogonal matrix $\mathbf{U}$ ($\mathbf{U}^T\mathbf{U} = \mathbf{I}_k$) are a basis for the column space of $\mathbf{Y}$. Thus, $\mathbf{U}$ represents the xyz-cube and can be identified with a point on the Grassmannian $G(k, n)$. Once the hyperspectral movie is mapped to a sequence of points on $G(k, n)$, the pairwise distances between these points may be found using an appropriate function of the angles between subspaces. Recall, for instance, that the chordal distance between k-dimensional subspaces $\mathcal{P}$ and $\mathcal{Q}$, is given by $d_c(\mathcal{P}, \mathcal{Q}) = \|\sin \boldsymbol{\theta}\|_2$, and the geodesic distance is $d_g(\mathcal{P}, \mathcal{Q}) = \|\boldsymbol{\theta}\|_2$, where $\boldsymbol{\theta}$ is the k-dimensional vector of the principal angles $\theta_i, i = 1, \ldots, k$, $0 \leq \theta_1 \leq \theta_2 \leq \ldots \leq \theta_k \leq \pi/2$, between $\mathcal{P}$ and $\mathcal{Q}$, see also Section 4.2.

In this study, we measure the similarity of two points with the smallest principal angle, $d_1 = \theta_{min} = \theta_1$, between the points [46, 55]. In fact, we observed in our experiments that using d_p resulted in stronger topological signals than did d_c and d_g. Once the sequence of cubes is mapped to $G(k, n)$, the matrix of all pairwise "distances" is computed, and we apply PH to generate $Betti_0$ barcodes to see the number of connected components (clusters) in the point cloud on the Grassmannian, corresponding to the raw HSI data.

5.4. Experimental Results

In this section, we show results obtained by the proposed approach applied to the detection of chemical signals in the collection of data cubes of the Long-Wavelength Infrared (LWIR) data set [40], see also Section 3.3.3. Recall that the LWIR data set is collected by an interferometer in the 8-11 μm range of the electromagnetic spectrum. During a single scanning, 256×256 pixel images are collected across 20 wavelengths within this range, forming a $256 \times 256 \times 20$ data cube. Here we consider a data collection event consisting of releasing a pre-determined quantity of Triethyl Phosphate (TEP) into the air to create an aerosol plume for detection against natural background. A series of 561 data cubes records the entire event from "pre-burst" to "post-burst", as a hyperspectral movie.

To strengthen topological signals, the experimental setting includes:

- dimension reduction of the band space using SSVM-based feature selection;
- finding the patch in the images that contains the chemical cloud;
- mapping selected (sub)cubes to the Grassmannian;
- computing the pairwise distances on the manifold using d_1;
- generating PH $Betti_0$ (or 0-dimensional) barcodes for clustering.

Here we use 3 (out of 20) wavelength bands 3,11, and 15 (Table 3.6) selected by Band Selection SSVM Algorithm 2 via classifying the TEP data pixels against the background pixels. A single wavelength of the data set in question is shown in Figure 5.9 for a given time in the movie, the image contains a plume that is not visible.

To validate our results, we determine the location of the chemical plume in the cubes using the adaptive-cosine-estimator (ACE) [64]. The ACE detector is one of the benchmark hyperspectral detection algorithms, that (geometrically) computes the squared cosine of the angle between the whitened test pixel and the whitened target's spectral signature. Based

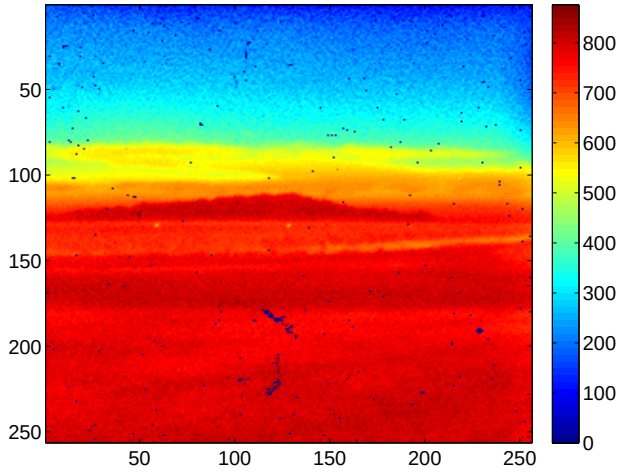
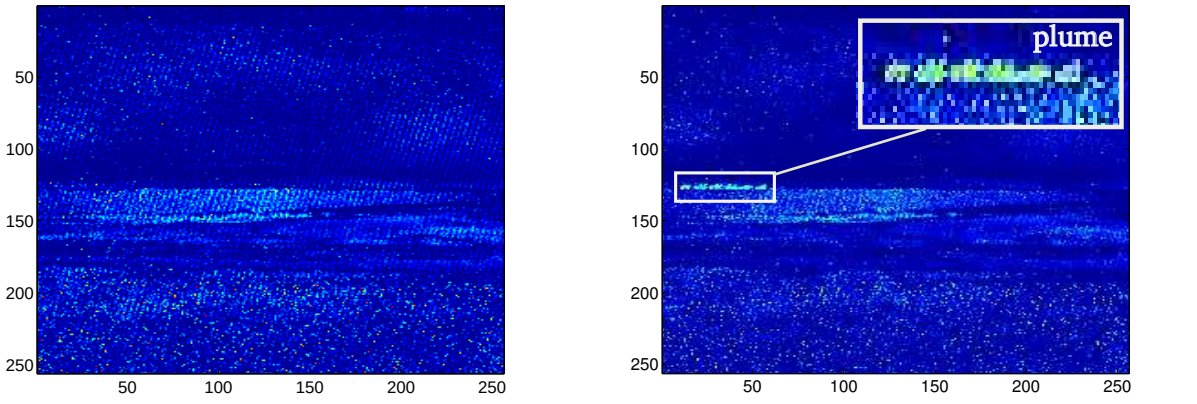


Figure 5.9: A single wavelength of an hyperspectral image containing a plume that is not visible. This is part of a cube drawn from the time dependent LWIR sequence of HSI cubes.

on a chosen threshold, this ACE score indicates if the chemical is present in the test pixel. Figure 5.10 shows two images corresponding to cube 111 without a plume, Figure 5.10a, and cube 113 with a chemical plume detected by the ACE, Figure 5.10b.



(a) (b)

Figure 5.10: The ACE detector application results on the LWIR data cubes: (a) the image of cube 111 with no plume detected; (b) the image of cube 113 with plume detected by the ACE and zoomed for better visualization.

5.4.1. Experiment on Subsets of Cubes. We first consider several small subsets of the set of total 561 TEP cubes and generate 0-dimensional barcodes under the Grassmannian framework. We analyze PH results on the following subsets:

(1) "pre-burst" cubes 104-111;

(2) "pre-burst" cubes 104-111 and TEP release cube 112, in which a chemical plume occurs for the first time in the HSI movie;

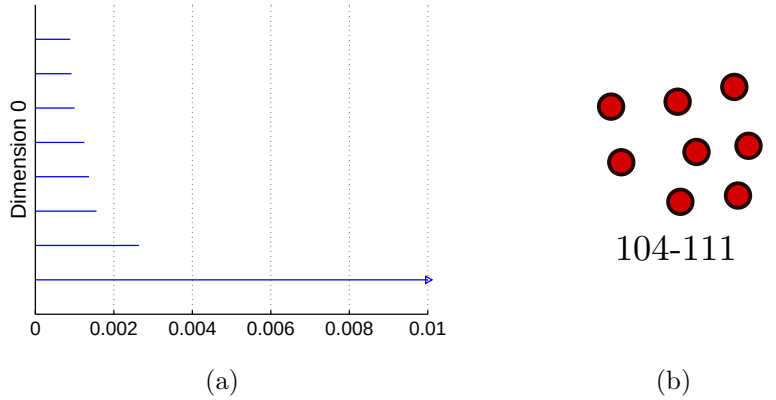
(3) "pre-burst" cubes 104-111 and cubes 112 and 114, both containing a TEP plume;

(4) "pre-burst" cubes 104-111 and cubes 112-116, containing an evolving TEP plume.

To generate $Betti_0$ barcodes on these subsets, a "plume location" patch of size $4 \times 8 \times 3$ from each cube is mapped to a point on $G(3, 4 \times 8) = G(3, 32)$, with "3" corresponding to the number of bands preselected. We use pixel rows 124 to 127 and pixel columns 34 to 41, as this patch size is close to the size of the plume detected by the ACE in the first "burst" cubes, such as 112 or 113.

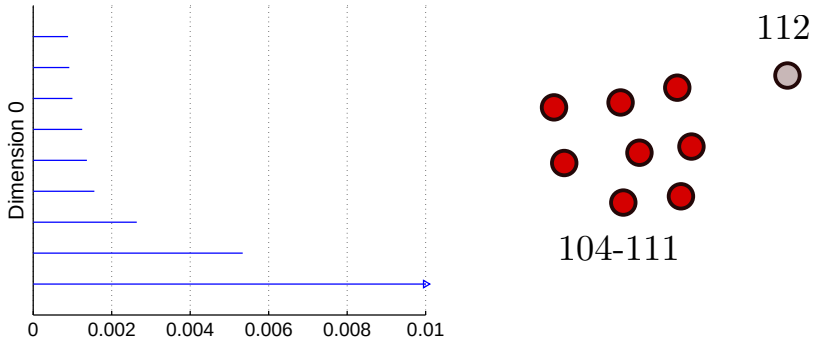
Let us consider PH 0-dimensional barcodes generated for the first three subsets. Recall that the longest horizontal bars in a barcode (i.e., persistent over many scales) correspond to the strongest topological signal and tell us about structure in a point cloud. In Figure 5.11, PH result on the "pre-burst" cubes 104-111 indicates that we basically have one cluster of points. Once we add at least one point containing a plume, the situation changes, see Figures 5.12 and 5.13. Here we observe formation of two (Figure 5.12) and three (Figure 5.13) connected components in corresponding subsets of points on $G(3, 32)$. In particular, at scale $\epsilon = 4 \times 10^{-3}$, all the three barcodes have different number of clusters, reflecting the situation before and after the release of TEP.

Let us now consider subset (4) of subsequent points 104-116 and PH clustering results over many scales, shown in Figure 5.14. The 0-dimensional barcode in Figure 5.14a has different



(a) (b)

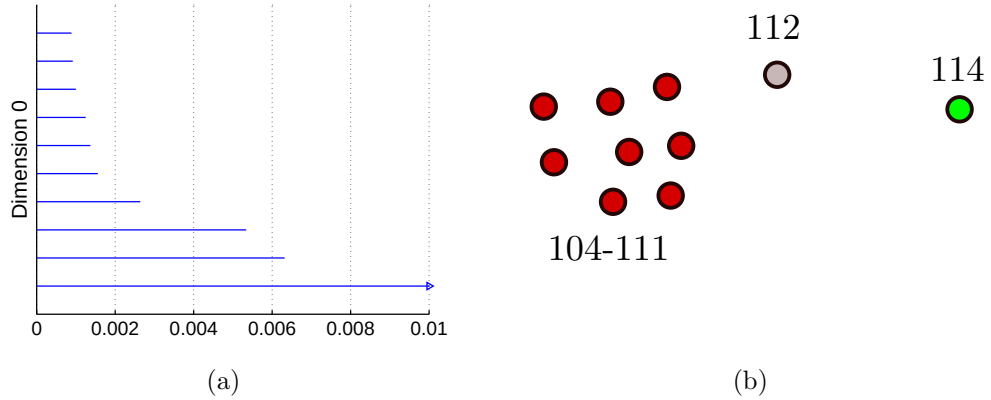
Figure 5.11: (a) $Betti_0$ barcode generated on points on $G(3, 32)$, corresponding to $4 \times 8 \times 3$ subcubes 104 to 111 (just before TEP release); (b) the cluster of points 104-111 on $G(3, 32)$ at $\epsilon = 4 \times 10^{-3}$.



(a) (b)

Figure 5.12: (a) $Betti_0$ barcode generated on points on $G(3, 32)$, corresponding to $4 \times 8 \times 3$ subcubes 104 to 111 (just before TEP release) and 112 (TEP release); (b) the cluster of points 104-111 (red) and isolated point 112 (gray) on $G(3, 32)$ at $\epsilon = 4 \times 10^{-3}$.

numbers of connected components as the scale parameter ϵ increases. For instance, at the small scale of $\epsilon = 5 \times 10^{-4}$, all the points are disconnected (13 bars are present), which is shown schematically in Figure 5.14b by distinct coloring for each point. Figure 5.14c depicts the clustering that occurs at $\epsilon = 4 \times 10^{-3}$. At this scale, we have 6 clusters, with one cluster containing all the "pre-burst" points 104-111 (shown in red) (compare to Figure 5.11), and 5 clusters each containing isolated plume points 112 to 116, indicated by distinct colors.



(a) (b)

Figure 5.13: (a) $Betti_0$ barcode generated on points on $G(3, 32)$, corresponding to $4 \times 8 \times 3$ subcubes 104 to 111 (just before TEP release) and 112, 114 (TEP release); (b) the cluster of points 104-111 (red) and isolated points 112 (gray) and 114 (green) on $G(3, 32)$ at $\epsilon = 4 \times 10^{-3}$.

Later, at $\epsilon = 6 \times 10^{-3}$, PH detects 3 clusters of points: plume points 112 and 113 join the cluster of points 104-111, and points 115 and 116 merge into a separate cluster, with point 114 staying isolated, see Figure 5.14d. This can be interpreted as follows: points 112 and 113, where the plume first develops, are closer to the "pre-plume" cluster on $G(3, 32)$ than the points 114, 115, 116, as the shape of the plume changes. In particular, PH tells us that the points within a cluster are more similar to each other on the manifold than to the points from a different cluster or to an isolated point. Note that when ϵ is large enough, all points in a barcode merge into a single connected component.

5.4.2. Experiment on All Cubes. This experiment includes generating $Betti_0$ barcodes using all 561 TEP cubes. Similar to the experiment in Section 5.4.1, we consider $4 \times 8 \times 3$ subcubes "cut out" from different areas in each image such as the top (sky), the middle (horizon, where the plume develops), and the bottom (ground) for left, right, and center regions, respectively. See Figure 5.15 for an illustration of left subcubes from the sky mapped to $G(3, 32)$.

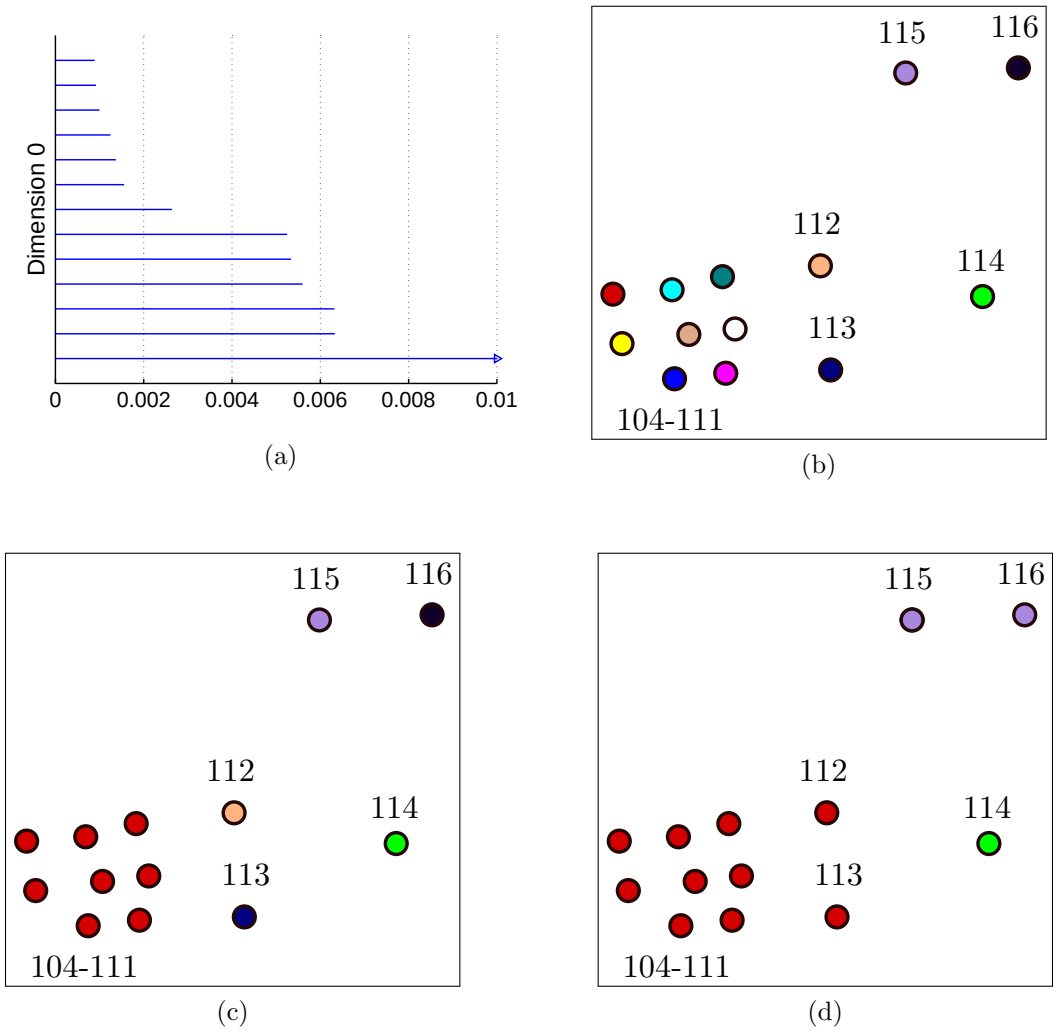


Figure 5.14: (a) $Betti_0$ barcode generated on points on $G(3, 32)$, corresponding to $4 \times 8 \times 3$ subcubes 104 to 116 selected from 561 TEP data cubes; (b) 13 isolated points 104-116 on $G(3, 32)$ at $\epsilon = 5 \times 10^{-4}$, shown by distinct colors; (c) 6 clusters at $\epsilon = 4 \times 10^{-3}$: the red colored cluster of points 104-111 and 5 isolated points 112-116, shown by distinct colors; (d) 3 clusters at $\epsilon = 6 \times 10^{-3}$: the cluster of points 104-113 (red), the isolated point 114 (green), and the cluster of points 115 and 116 (purple).

We generate nine 0-dimensional barcodes for the different regions described above, see Figure 5.16. Notice the similarity of the barcodes along the first (sky) and third (ground) rows, indicating uniformity in these regions throughout the hyperspectral movie. In contrast, the plume occurs and develops along the horizon. This dynamic movement within the scene is reflected in the fluctuation of the barcodes, see the second row in Figure 5.16.

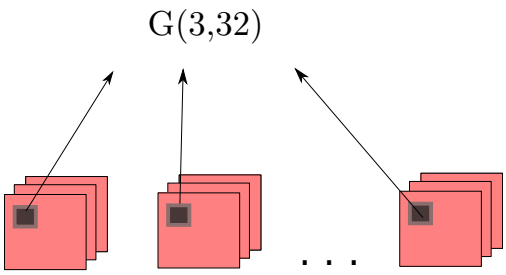


Figure 5.15: Grassmannian setting for the 561 top (sky) left $4 \times 8 \times 3$ subcubes.

Let us further consider the clusters forming in the 0-dimensional barcode in Figure 5.16d. This barcode is generated from the 561 points corresponding to the left horizon $4 \times 8 \times 3$ region in each data cube limited by pixel rows 124 to 127 and pixel columns 34 to 41. This region belongs to the plume formation area, as detected by the ACE for cube 112. Figure 5.17 shows a detailed (zoomed) version of the barcode in Figure 5.16d.

At scale $\epsilon = 1.5 \times 10^{-3}$, there are 31 bars corresponding to 31 connected components on $G(3, 32)$, with 28 isolated points from frames 111 to 142, one cluster containing frames 134, 135, and 137, one cluster containing frame 519, and another containing all other frames. At scale $\epsilon = 2 \times 10^{-3}$, we have 19 bars corresponding to 19 connected components on $G(3, 32)$, with 18 isolated frames from 112 to 129, and one cluster containing all the rest. Note that these bars persist for a large range of parameter value (to just beyond 3×10^{-3}), indicating a large degree of separation. At $\epsilon = 4 \times 10^{-3}$, we have 13 clusters with 11 isolated frames 112, 114-118, 120-123, and 125, one cluster of frames 119 and 124, and the other one containing everything else.

Note that cubes following frame 111 are where the plume first occurs with the highest concentration of chemical and changes very fast. PH detects separation of these points from pre-plume cubes at multiple scales. The Grassmannian framework together with PH treats

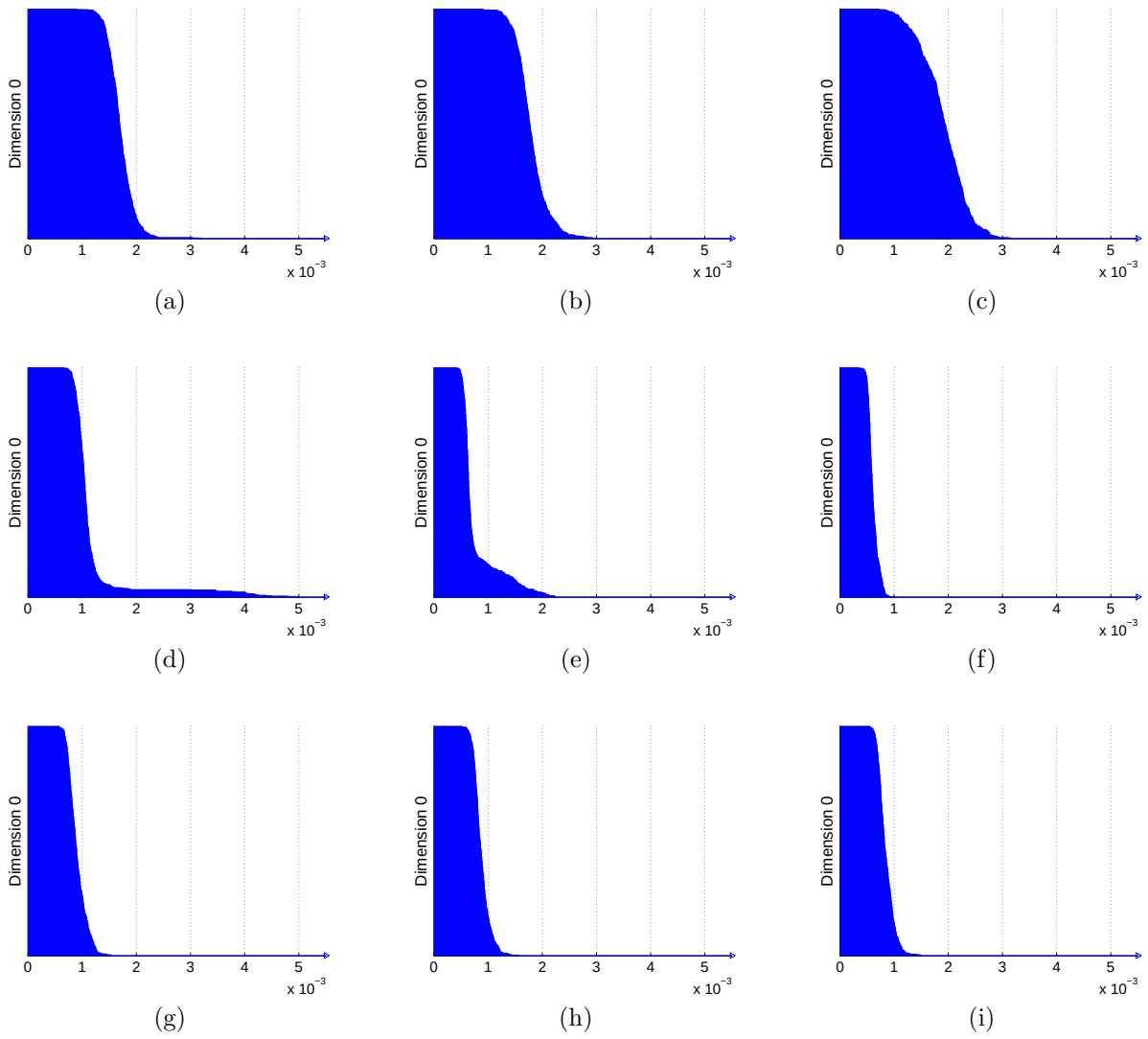


Figure 5.16: $Betti_0$ barcodes generated on selected $4 \times 8 \times 3$ regions through all 561 TEP cubes, mapped to $G(3, 32)$: (a) top left; (b) top middle; (c) top right; (d) middle left; (e) center; (f) middle right; (g) bottom left; (h) bottom middle; (i) bottom right.

these points as far away from each other and from the rest of the points, therefore capturing the dynamics in the sequence of HSI images containing the chemical.

For the last experiment in this chapter, we consider clusters generated by PH on 561 points on $G(3, 32)$ corresponding to a horizon region located to the right from the plume area (as detected by the ACE in cube 112). We use pixel rows 124 to 127 and pixel columns 75 to 82 to create a patch of size $4 \times 8 \times 3$. Figure 5.18 contains the 0-dimensional barcode

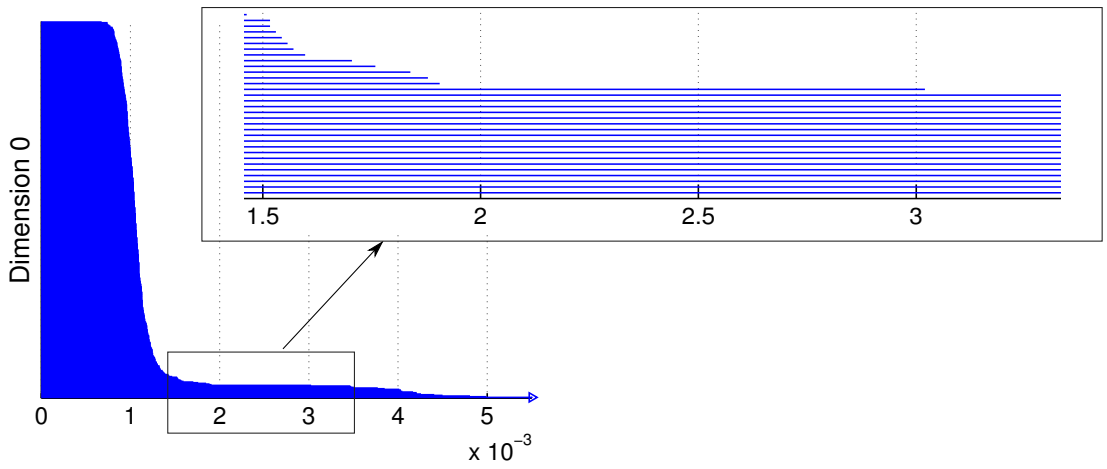


Figure 5.17: $Betti_0$ barcode generated on $4 \times 8 \times 3$ left horizon (plume formation) region limited by pixel rows 124-127 and columns 34-41, through all 561 TEP cubes, mapped to $G(3, 32)$.

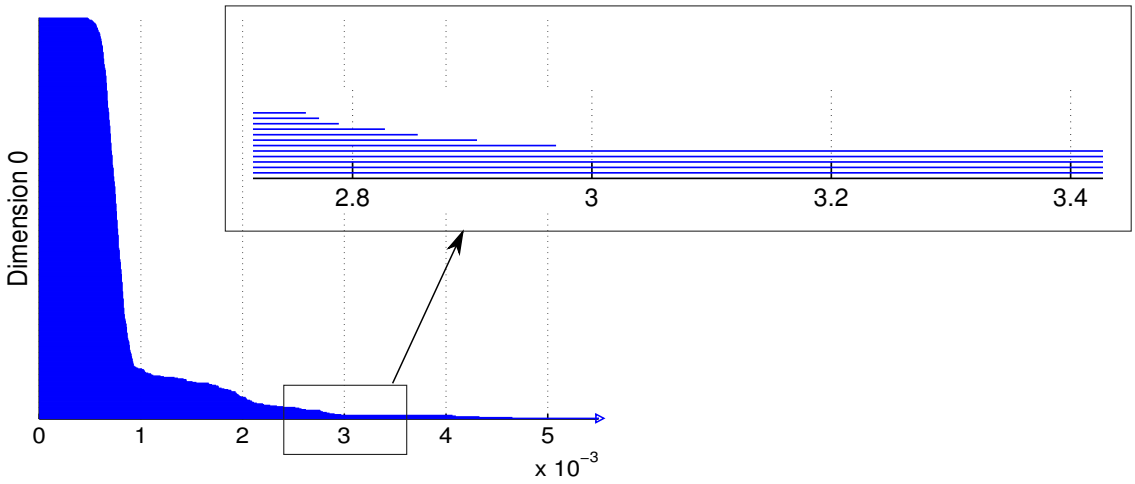


Figure 5.18: $Betti_0$ barcode generated on $4 \times 8 \times 3$ horizon region limited by pixel rows 124-127 and columns 75-82, through all 561 TEP cubes, mapped to $G(3, 32)$.

and its zoomed part. Analyzing connected components as ϵ varies, we observe that they differ from those found in the previous all-cubes experiment, see Figure 5.17. At scale $\epsilon = 1.5 \times 10^{-3}$, we have 52 connected components on $G(3, 32)$ corresponding to 47 isolated points from 119 to 141, 145 to 165, and 170 to 172. The other points are connected into 4 smaller clusters (142,143,144), (166,167), (168,169), and (173,174), and one big cluster

containing all the other points. At scale $\epsilon = 2 \times 10^{-3}$, there are 30 connected components on the Grassmannian, including 25 isolated points from 119 to 127, 129 to 140, 151 to 156, and 149. The clusters are (128,136-138), (141-150), (157,158), (162-164), and one cluster containing all the rest. Further, at scale $\epsilon = 3 \times 10^{-3}$, the barcode plot has 5 bars that persist over a large range of values, namely, up to a little beyond 4×10^{-3}: 4 isolated points from frame 121 to 124 and one cluster containing all the rest.

We observe that for this region, PH separates points from frame 119 and later, in contrast to the frames separated from frame 112 in the left horizon region experiment (Figure 5.17). Note that points 112 to 118 are "plume-free" as the plume does not reach this region until frame 119. It is also interesting to note that points corresponding to frames 121 to 124 are kept isolated for a large range of scales, i.e., they are far away from each other and the rest of the points. PH (under the Grassmannian framework) treats these frames as experiencing the most significant changes in this region.

5.5. Summary

In summary, we presented a geometric framework for characterizing information in hyperspectral data cubes evolving in time. Persistent homology was employed to aid in detecting changes in topological structure on point clouds generated from raw HSI data under the Grassmannian framework. We observed that, depending on the PH parameter value ϵ, both all-cubes and subset-of-cubes experiments resulted in clustering that reflected the dynamical changes in the HSI sequences of cubes of the LWIR data set.

In the first experiment, with small subsets of Triethyl Phosphate cubes mapped to the Grassmannian, PH $Betti_0$ barcodes captured the evolution of the plume when it first occurred and started evolving. In the second, all-cubes experiment, different regions of the cubes were

mapped to a manifold to generate barcodes. We observed changes in the barcode profiles obtained along the horizon ("plume") line, while the other regions in the cubes resulted in similar plots. Based on clustering results for the left horizon subcubes, several frames with a plume were treated by PH as isolated points on the manifold, in contrast to "pre-burst" points and points long after the release time, all clustered together. By comparing two horizon regions in the hyperspectral movie, we observed that for the same parameter ϵ values, persistence homology detected different subsets of frames, treated as isolated points on the manifold, therefore indicating the changes in the shape and location of the plume.

Having found these results promising, further research can be done to strengthen the topological signal. We are working to employ other mappings, other (pseudo)metrics on the Grassmannian, and $Betti_1$ barcodes. We are further making a comparative analysis of no-plume and plume data cubes, based on mapping subsets of pixels to $G(1, n)$ where n is the total number of spectral bands.

CHAPTER 6

Conclusion

In this dissertation, we developed novel algorithmic frameworks for embedded feature selection and pattern recognition on Grassmannians. Tools from geometry, topology, optimization, and machine learning can be effectively used for exploring geometric structure and for constructing relationships in data. For the illustration of our approaches, we presented experimental results obtained for some real-world applications. We particularly made the following contributions.

In Chapter 3, we proposed solving the hyperspectral band selection problem by using sparse linear SVMs. The supervised embedded approach exploits the sparsity promoting property of SSVMs to suppress features that do not contribute into classification process and, as a result, to reduce the band space dimension, keeping the most discriminatory features only. Our method includes bootstrap aggregating (bagging) for robustness, a new ratio-based elimination step for feature selection, the use of primal dual interior point solver for the SSVM (described in Chapter 2), and multiclass case extension. The proposed technique is effective and can be used in combination with other feature selection approaches.

In Chapter 4, we performed set-to-set pattern recognition via classification of data on embedded Grassmannians. Multiple observations from a data class, organized as subspaces on an abstract manifold, capture the signal variability of the class and lead to better prediction rates. Multidimensional scaling provides a low-dimensional embedding of the manifold into Euclidean space, preserving or approximating the geometry of the Grassmannian, depending on the choice of a distance metric on the manifold. In particular, the chordal distance framework resulted in isometric embeddings. This approach allows for application of any

classification technique in the embedding Euclidean space. We apply SSVMs for classification and identification of optimal dimensions of embedded subspaces. We observed that, under the smallest principal angle pseudometric framework, classification accuracy grew up to 100% even in high difficulty binary classification cases, and only one dimension of the embedding space was needed to separate the classes. To expand the use of the method, future work will include comparison of Grassmannians $G(k, n)$ and $G(l, n)$, $k \neq l$, and in particular, the case when $k > l = 1$.

Finally, in Chapter 5, we applied persistent homology (PH) to the analysis of hyperspectral movies. The Grassmannian framework, used to organize large volumes of hyperspectral data, afforded a form of data compression while retaining pertinent structure. Particularly, sequences of subcubes from different time frames in the LWIR hyperspectral movie were mapped to a Grassmann manifold, forming point clouds for analysis. Persistent homology was used to determine and analyze connected components (clusters) in the point clouds, based on the pairwise distances between the points and 0-dimensional holes that persisted over the large range of scales. The use of the smallest principal angle as a distance measure on the manifold provided strong topological signals. PH clustering results were different for the regions in the movie that contained the evolving chemical plume. In particular, PH captured the dynamics of the plume along the horizon line by treating the plume containing points as isolated components that were far away from each other and from the non-plume points. Future work will include exploration of different settings in combination with higher-dimensional persistent barcodes which may provide interesting opportunities for expansion and new applications of this approach.

Bibliography

[1] D. Landgrebe, "Hyperspectral image data analysis," *Signal Processing Magazine, IEEE*, vol. 19, no. 1, pp. 17–28, 2002.

[2] M. Kirby, *Geometric Data Analysis: An empirical Approach to Dimensionality Reduction and the Study of Patterns*. New York: John Wiley and Sons, Inc., 2001.

[3] M. Anderle, D. R. Hundley, and M. J. Kirby, "The Bilipschitz criterion for dimension reduction mapping design," *Intelligent Data Analysis*, pp. 85–104, 2002.

[4] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[5] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[7] S. Chepushtanova, C. Gittins, and M. Kirby, "Band selection in hyperspectral imagery using sparse support vector machines," in *Proceedings SPIE DSS 2014*, vol. 9088, pp. 90881F–90881F15, 2014.

[8] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[9] A. Zomorodian and G. Carlsson, "Computing persistent homology," *Discrete & Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005.

[10] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pp. 454–463, 2000.

[11] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[12] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive computation and machine learning, MIT Press, 2002.

[13] B. Schölkopf, K. Tsuda, and J. Vert, *Kernel Methods in Computational Biology*. Computational Molecular Biology, MIT Press, 2004.

[14] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[15] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, pp. 1778–1790, Aug. 2004.

[16] L. Zhang and W. Zhou, "On the sparseness of 1-norm support vector machines," *Neural networks : the official journal of the International Neural Network Society*, vol. 23, pp. 373–85, Apr. 2010.

[17] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," in *Machine Learning Proceedings of the Fifteenth International Conference*, ICML '98, pp. 82–90, Morgan Kaufmann, 1998.

[18] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," *Neural Information Processing Systems*, no. 16, 2003.

[19] O. L. Mangasarian and G. Kou, "Feature selection for nonlinear kernel support vector machines," pp. 231–236, Oct. 2007.

[20] D. Zhou, B. Xiao, H. Zhou, and R. Dai, "Global geometry of svm classifiers," tech. rep., June 2002.

[21] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*. Athena Scientific, 1st ed., 1997.

[22] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*. Springer, 3nd ed., 2008.

[23] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, 2006.

[24] N. Keshava, "Distance metrics and band selection in hyperspectral processing with applications to material identification and spectral libraries," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 7, pp. 1552–1565, 2004.

[25] B. Guo and S. R. Gunn, "Band selection for hyperspectral image classification using mutual information," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, pp. 522–526, 2006.

[26] A. Martinez-Uso, F. Pla, P. Garcia-Sevilla, and J. M. Sotoca, "Automatic band selection in multispectral images using mutual information-based clustering," in *Proceedings of the 11th Iberoamerican Congress in Pattern Recognition*, pp. 644–654, 2006.

[27] A. Martinez-Uso, F. Pla, J. M. Sotoca, and P. Garcia-Sevilla, "Clustering-based hyperspectral band selection using information measures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 4158–4171, 2007.

[28] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1, pp. 273–324, 1997.

[29] S. Li, H. Wu, D. Wan, and J. Zhu, "An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine," *Knowledge-Based Systems*, vol. 24, no. 1, pp. 40–48, 2011.

[30] T. N. Lal, O. Chapelle, J. Weston, and A. Elisseeff, "Embedded methods," in *Feature Extraction*, pp. 137–165, Springer Berlin Heidelberg, 2006.

[31] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[32] R. Archibald and G. Fann, "Feature selection and classification of hyperspectral images with support vector machines," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 4, pp. 2007–2010, 2007.

[33] R. Zhang and J. Ma, "Feature selection for hyperspectral data based on recursive support vector machines," *International Journal of Remote Sensing*, vol. 30, pp. 3669–3677, July 2009.

[34] G. Fung and O. Mangasarian, "A feature selection Newton method for support vector machine classification," *Computational Optimization and Applications*, vol. 28, no. 2, pp. 185–202, 2004.

[35] J. Bi, K. P. Bennett, M. Embrechts, B. C. M., and M. Song, "Dimensionality reduction via sparse support vector machines," *The Journal of Machine Learning Research*, vol. 3, pp. 1229–1243, 2003.

[36] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[37] H. Zou, "An improved 1-norm SVM for simultaneous classification and variable selection," in *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, no. 2001, pp. 675–681, 2007.

[38] "Airborne Visible Infrared Imaging Spectrometer AVIRIS Indian Pines data set." Retrieved July 10, 2013 from `https://engineering.purdue.edu/~biehl/MultiSpec`.

[39] Y. Liu and Y. Zheng, "One-against-all multi-class SVM classification using reliability measures," in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN) 2005*, vol. 2, pp. 849–854, July 2005.

[40] B. R. Cosofret, D. Konno, A. Faghfouri, H. S. Kindle, C. M. Gittins, M. L. Finson, T. E. Janov, M. J. Levreault, R. K. Miyashiro, and W. J. Marinelli, "Imaging sensor constellation for tomographic chemical cloud mapping," *Applied Optics*, vol. 48, pp. 1837–1852, 2009.

[41] A. Martinez-Uso, "Band selection tool." `http://www.vision.uji.es/~adolfo/BandSelectionTool_files/BandSelectionTool.htm`.

[42] A. Zare and P. Gader, "Hyperspectral band selection and endmember detection using sparsity promoting priors," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, pp. 256–260, Apr. 2008.

[43] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010.

[44] L. B. Ziegelmeier, *Exploiting Geometry, Topology and Optimization for Knowledge Discovery in Big Data*. PhD thesis, Colorado State University, Fort Collins, May 2013.

[45] P.-A. Absil, R. Mahony, and R. Sepulchre, "Riemannian geometry of grassmann manifolds with a view on algorithmic computation," *Acta Applicandae Mathematica*, vol. 80, no. 2, pp. 199–220, 2004.

[46] J.-M. Chang, M. Kirby, H. Kley, C. Peterson, B. A. Draper, and J. Beveridge, "Recognition of digital images of the human face at ultra low resolution via illumination spaces," *Computer Vision - ACCV 2007*, vol. 4884, pp. 733–743, 2007.

[47] J. Nash, "The imbedding problem for riemannian manifolds," *Annals of Mathematics*, vol. 63, no. 1, pp. 20–63, 1956.

[48] P. L. Robinson, "The sphere is not flat," *The American Mathematical Monthly*, vol. 113, no. 2, pp. 171–173, 2006.

[49] J. H. Conway, R. H. Hardin, and N. J. A. Sloane, "Packing lines, planes, etc.: packings in Grassmannian spaces," *Experimental Mathematics*, vol. 5, pp. 139–159, 1996.

[50] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate analysis*. Academic Press, 1979.

[51] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[52] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 90, pp. 2323–2326, 2000.

[53] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[54] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina, "Exploiting manifold geometry in hyperspectral imagery," *IEEE Transactions Geoscience Remote Sensing*, vol. 43, no. 3, pp. 441–454, 2005.

[55] S. Chepushtanova and M. Kirby, "Classification of hyperspectral imagery on embedded Grassmannians," in *IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS 2014)*, June 2014.

[56] Å. Björck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Mathematics of Computation*, vol. 27, no. 123, pp. 579–594, 1973.

[57] T. F. Cox and M. Cox, *Multidimensional Scaling*. Chapman and Hall, 2 ed., 2000.

[58] "Reflective Optics Spectrographic Imaging System ROSIS Pavia University data set." Retrieved July 2, 2014, from `http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes`.

[59] S. Chepushtanova, M. Kirby, C. Peterson, and L. Ziegelmeier, "An application of persistent homology on Grassmann manifolds for the detection of signals in hyperspectral imagery," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2015*, July 2015.

[60] V. Farley, A. Vallières, A. Villemaire, M. Chamberland, P. Lagueux, and J. Giroux, "Chemical agent detection and identification with a hyperspectral imaging infrared sensor," in *SPIE Defense, Security, and Sensing*, vol. 6739, pp. 673918–673918–12, 2007.

[61] R. Ghrist, "Barcodes: The persistent topology of data," *Bulletin of the American Mathematical Society*, vol. 45, pp. 61–75, 2008.

[62] A. Hatcher, *Algebraic topology*. Cambridge, New York: Cambridge University Press, 2002.

[63] A. Tausz, M. Vejdemo-Johansson, and H. Adams, "JavaPlex: A research software package for persistent cohomology," in *Proceedings of ICMS 2014* (H. Hong and C. Yap, eds.), Lecture Notes in Computer Science 8592, pp. 129–136, 2014.

[64] D. Manolakis, "Signal processing algorithms for hyperspectral remote sensing of chemical plumes," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008 (ICASSP 2008)*, pp. 1857–1860, March 2008.